

**IS jednoduché účetnictví jako
vzorový příklad do výuky**

**Single-entry bookkeeping
application course's as sample**

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 03.05.2011

.....

Rád bych na tomto místě poděkoval své vedoucí za vedení, užitečné rady a připomínky, které směřovaly ke zlepšení diplomové práce.

Abstrakt

Cílem diplomové práce je navrhnout a naimplementovat IS daňové evidence, použitelné pro menší firmy, které vedou své účetnictví systémem daňové evidence. Část informačního systému je zpracována jako vzorový příklad pro použití v databázových předmětech navazujícího studia. Součástí diplomové práce je vypracovaná specifikace požadavků, které jsou podrobně zanalyzovány, je proveden návrh implementace a následně i samotná implementace. Součástí implementace je i otestování funkcionality informačního systému. Pro programátory a cílovou skupinu uživatelů byly zpracovány příručky. Informační systém je naimplementován v jazyce PHP a jako úložiště dat byla použita relační databáze MySQL. Při implementaci byly použity technologie Smarty, Ajax, JS, jQuery.

Klíčová slova: daňová evidence, jednoduché účetnictví, informační systém, faktura

Abstract

The aim of this thesis is to design and implement an information system for single-entry bookkeeping, which is applicable to small companies. Part of the information system is designed in a way that serves as a tutorial that can be used in the Master's database systems courses. The thesis discusses many aspects of the software development lifecycle: specification of requirements, analysis, implementation design, the implementation itself and also testing of the system's functionality. Manuals were created for both users and developers. The system is implemented as a web application in PHP with MySQL as a database management system. Many other technologies, such as Smarty, AJAX and jQuery were used in the implementation phase.

Keywords: single-entry bookkeeping, Information System, invoice

Seznam použitých zkratk a symbolů

IS	– Informační systém
DE	– Daňová evidence
SW	– Software
OS	– Operační systém
DB	– Databáze
DPH	– Daň z přidané hodnoty
GUI	– Graphical User Interface
ID	– Identification
OK	– All correct
CD	– Compact Disc
IČ	– Identifikační číslo ekonomického subjektu
DIČ	– Daňové identifikační číslo
XSS	– Cross-site scripting
JS	– JavaScript
JSON	– JavaScript Object Notation
ER-diagram	– Entity Relationship Diagram
DFD	– Data Flow Diagram
STD	– State Transition Diagram
SQL	– Structured Query Language
PHP	– Hypertext preprocessor
MySQL	– My Structured Query Language
XHTML	– eXtensible HyperText Markup Language
CSS	– Cascading Style Sheets
AJAX	– Asynchronous Javascript technology And XML

Obsah

1 Úvod	6
2 Stanovení a popis cílů	7
3 Daňová evidence	9
3.1 Srovnání daňové evidence a účetnictví	9
3.2 Srovnání existujících účetních programů	10
4 Analýza	14
4.1 Specifikace požadavků	14
4.1.1 Vstupy aplikace	14
4.1.2 Výstupy aplikace	19
4.1.3 Funkce	20
4.1.4 Okolí systému	21
4.1.5 Bezpečnost aplikace	21
4.1.6 Nefunkční požadavky	22
4.2 Datová analýza	22
4.2.1 ER diagram - evidence faktur	22
4.2.2 Lineární zápis typů entit	23
4.2.3 Datový slovník	27
4.3 Funkční analýza	29
4.3.1 Kontextový diagram	29
4.3.2 Diagram datových toků (DFD)	30
4.3.3 Minispecifikace	34
4.3.4 Dynamická analýza	38
4.4 Návrh implementace	39
4.4.1 Transakční analýza	39
4.4.2 Analýza zálohování dat	43
4.4.3 Analýza archivace dat	45
4.4.4 Návrh grafického uživatelského rozhraní (GUI)	45
5 Realizace IS daňové evidence	47
5.1 Popis modulů IS	47
5.2 Rozbor modulu faktura	48
5.2.1 Interaktivní seznam faktur	49
5.2.2 Přepočet cen ve formuláři	51
5.2.3 Generování tiskových sestav do PDF	52
6 Protokol testování	55
6.1 Testování informačního systému	55
7 Ukázkové příklady pro výuku	57

8 Závěr	59
9 Reference	61
10 Obsah CD	63

Seznam tabulek

1	Srovnání funkcionality účetních programů POHODA Mini a Money S3 Mini	13
2	Datový slovník	29
3	Návrh četnosti záloh dat - tabulek databáze	45
4	Testování informačního systému	56

Seznam obrázků

1	Money S3 START - ukázka GUI	11
2	POHODA START - ukázka GUI	12
3	Ukázka části ER diagramu - evidence faktur	23
4	Kontextový diagram	30
5	DFD diagram 0. úrovně	31
6	DFD diagram 1. úrovně - evidence faktur I	32
7	DFD diagram 1. úrovně - evidence faktur II	33
8	DFD diagram 1. úrovně - evidence faktur III	34
9	Minispecifikace - přidání faktury	35
10	Ukázka formuláře - přidání faktury	35
11	STD diagram - vystavená faktura	39
12	Návrh GUI pro evidenci faktur	46
13	Část formuláře pro přepočet cen faktury	52
14	Ukázka vygenerované vystavené faktury do PDF souboru	54
15	Náhled na GUI části adresáře - evidence firem	57

Seznam výpisů zdrojového kódu

1	SQL dotaz pro načtení seznamu faktur	49
---	--	----

1 Úvod

Téma diplomové práce se zabývá vytvořením informačního systému pro vedení daňové evidence (dříve označované jako jednoduché účetnictví). Daňová evidence je jedním ze dvou dostupných způsobů, kterým může účetní jednotka vést účetní agendu. Účetní jednotkou se rozumí osoba zodpovědná za vedení účetní agendy, tedy daňové evidence nebo účetnictví. V současné době je však jako účetní jednotka mnohdy označována pouze osoba zodpovědná za vedení klasického účetnictví.

Existuje celá řada aplikací a nástrojů, které mohou sloužit účetní jednotce k vedení daňové evidence. Aplikace se od sebe liší nejen uživatelským rozhranním, ale v mnohých případech také funkcionalitou. Ve většině případů se jedná o soubor firemně vyspělých konkurentů, kteří se snaží zaujmout uživatele co nejlepšími poměry ceny, funkcionality a použitelnosti, přičemž skloubit tyto aspekty s přívětivostí uživatelského rozhraní není příliš jednoduché.

Tato práce nemá za úkol vyrovnat se svou funkcionalitou a kvalitou existujícím aplikacím, které jsou vlastnictvím velkých firem a dostupné na trhu již řadu let, ale bere si za cíl dokázat, že lze vytvořit jednoduchý účetní program, který bude dodržovat platné zákony o účetnictví a zároveň bude pro účetní jednotky použitelný a ulehčí tak práci těm účetním jednotkám, které vedou účetní agendu formou daňové evidence.

Realizace informačního systému proběhne od počátečního zadání, přes analýzu a návrh implementace až k samotné implementaci. Dále musí proběhnout nutná testování a vytvoření příruček pro uživatele a programátory. Informační systém by měl být navržen tak, aby nebylo příliš složité jej v případě změn účetních zákonů upravit anebo jej případně rozšířit o další funkcionalitu. Část informačního systému bude zpracována jako vzorový příklad pro výuku databázových předmětů navazujícího studia.

2 Stanovení a popis cílů

Hlavním cílem diplomové práce je vytvořit informační systém pro účetní jednotky, které pro vedení účetní agendy využívají jako systém účtování daňovou evidenci. Práce bude vypracována po jednotlivých krocích, od podrobné specifikace zadání až po implementaci a dokumentaci. Část IS (informačního systému) má být zpracována jako vzorový příklad pro použití v databázových předmětech navazujícího studia.

Jednotlivé kroky pro vytvoření IS:

- Zjistěte požadavky na vytvoření IS pro vedení daňové evidence (jednoduchého účetnictví) dle platných zákonů o daňové evidenci.
- Vypracujte datovou, funkční i časovou analýzu dle těchto požadavků.
- Vypracujte návrh implementace pro vybrané vhodné, moderní technologie (objektově orientovaný model, AJAX, jQuery, PHP, MySQL).
- Navržený IS implementujte a otestujte.
- Napište programátorskou a uživatelskou příručku.

Prvním cílem je stanovit požadavky nutné k vytvoření informačního systému. Při stanovení požadavků na IS je především nutné striktně dodržet omezení, dané aktuálně platnými zákony. Požadavky na samotný IS daňové evidence byly před počátkem realizace stanoveny zadavatelem a v těchto požadavcích bylo uvedeno, že součástí IS budou následující moduly:

- peněžní deník s přípravou vedení ve 2 měnách (Euro)
- zálohování účetnictví
- evidence faktur, objednávek a nabídek
- evidence zaměstnanců, vozidel, majetku
- roční uzávěrky
- adresář, kalendář
- vedení skladu
- a další potřebné evidence

Po zjištění a zpracování potřebných požadavků bude vytvořena analýza k danému IS, která podrobněji rozvede každý ze stanovených požadavků. Zpracovány budou 3 druhy analýzy. Datová analýza pro zjištění základních informací o použitém databázovém modelu a dále funkční a časová analýza - zjištění podrobností o funkcionalitě a životních cyklech jednotlivých procesů.

Návrhem implementace budou upřesněny parametry analýzy, proběhnou další analýzy potřebné k implementaci informačního systému, stanoví se podrobnosti o použitých technologiích a zohlední se případné změny oproti původní analýze. Po fázi návrhu implementace již bude informační systém připraven k implementaci a vznikne i jednoduchý návrh designu aplikace.

Implementace IS proběhne podle návrhu implementace a finální verze analýzy. V průběhu implementace bude probíhat souběžně i testování funkcionality. Následně bude aplikace znovu otestována a bude zjištěno, zda aplikace odpovídá stanovenému zadání. Při testování bude zohledněno mnoho dalších aspektů, jakými jsou bezpečnost, validita jednotlivých stránek, schopnost odhalit chybu uživatele a zátěžový test aplikace, jejích skriptů a databáze.

Pro zjednodušení práce uživatelů - účetních jednotek - které s informačním systémem budou pracovat, bude vytvořena uživatelská příručka, obsahující popis veškeré práce uživatele se systémem včetně vysvětlení zkratk, pojmů, ikon, atd. Primárním cílem vytvořeného manuálu by tak mělo být ulehčení zejména počátečních prací uživatelů se systémem. Součástí řešení informačního systému bude i programátorská příručka, zahrnující popis tříd, metod a funkcí jednotlivých souborů a skriptů. Také kód by měl být vhodně okomentován pro další možné budoucí rozšíření a pro jednodušší použití ve výuce databázových předmětů.

3 Daňová evidence

V současné době existují podle platných zákonů o účetnictví dva způsoby, jakými lze vést účetní agendu:

- daňová evidence (dříve jednoduché účetnictví)
- účetnictví (dříve podvojně účetnictví)

3.1 Srovnání daňové evidence a účetnictví

Zákon o účetnictví (č. 563/1991 §1) stanovuje, kdo vede účetnictví. Jedná se o účetní jednotky, kterými jsou (následující seznam bodů je přesnou citací zákona o účetnictví [1]):

- právnické osoby se sídlem v ČR
- fyzické osoby - podnikatelé zapsaní v obchodním rejstříku
- ostatní fyzické osoby, jejichž obrat přesáhl za bezprostředně předcházející kalendářní rok 25 milionů Kč
- ostatní fyzické osoby, které vedou účetnictví na základě svého rozhodnutí
- ostatní fyzické osoby - podnikatelé a zároveň účastníci sdružení bez právní subjektivity, přičemž alespoň 1 účastník tohoto sdružení vede účetnictví
- zahraniční osoby (podnikající v ČR)
- organizační složky státu
- ostatní fyzické osoby, kterým to ukládá zvláštní právní předpis

Daňovou evidenci tedy naopak smí vést pouze fyzická osoba, která nesplňuje žádný z výše uvedených bodů. Jak daňová evidence, tak i účetnictví se musí řídit platnými zákony ČR, nicméně nároky na složitost vedení daňové evidence jsou o mnoho menší, než nároky na vedení účetnictví.

Daňová evidence slouží ke stanovení základu daně a zejména také ke stanovení výše daně z příjmů fyzických osob, které splňují podmínky pro vedení účetní evidence systémem daňové evidence. Podnikatel jako fyzická osoba musí vést údaje o příjmech a výdajích a o majetku a závazcích. Daňová evidence nahradila jednoduché účetnictví, které bylo zrušeno. Daňová evidence se řídí §7b zákona o daních z příjmů [2].

Na rozdíl od klasického účetnictví, zákon není ve vedení daňové evidence tak striktní a neurčuje přímo ani podobu, ve které má být daňová evidence vedena. Evidence může být vedena v jednoduché tištěné podobě a ve výjimečných případech bývá v praxi vedena i v programech, které nejsou primárně k daňové evidenci určeny, jako například Microsoft Excel. Se zrušením jednoduchého účetnictví se nastupující daňová evidence v některých ohledech dokonce i mírně zjednodušila. Spolu s evidencí již nemusí být veden například peněžní deník nebo majtkové karty.

Poplatník daně je povinen na základě daňové evidence zjistit základ daně z příjmu. Daňová evidence musí obsahovat potřebné údaje o (následující body jsou přesnou citací §7b zákona o dani z příjmu [3]):

- příjmech a výdajích, v členění potřebném pro zjištění základu daně
- majetku a závazcích

Pro vymezení hranic obsahu jednotlivých složek majetku jsou použity části zákona §6 až §9 vyhlášky č. 500/2002 Sb. Všechny existující daňové výdaje musí být při případné kontrole doložitelné potřebnými účetními doklady.

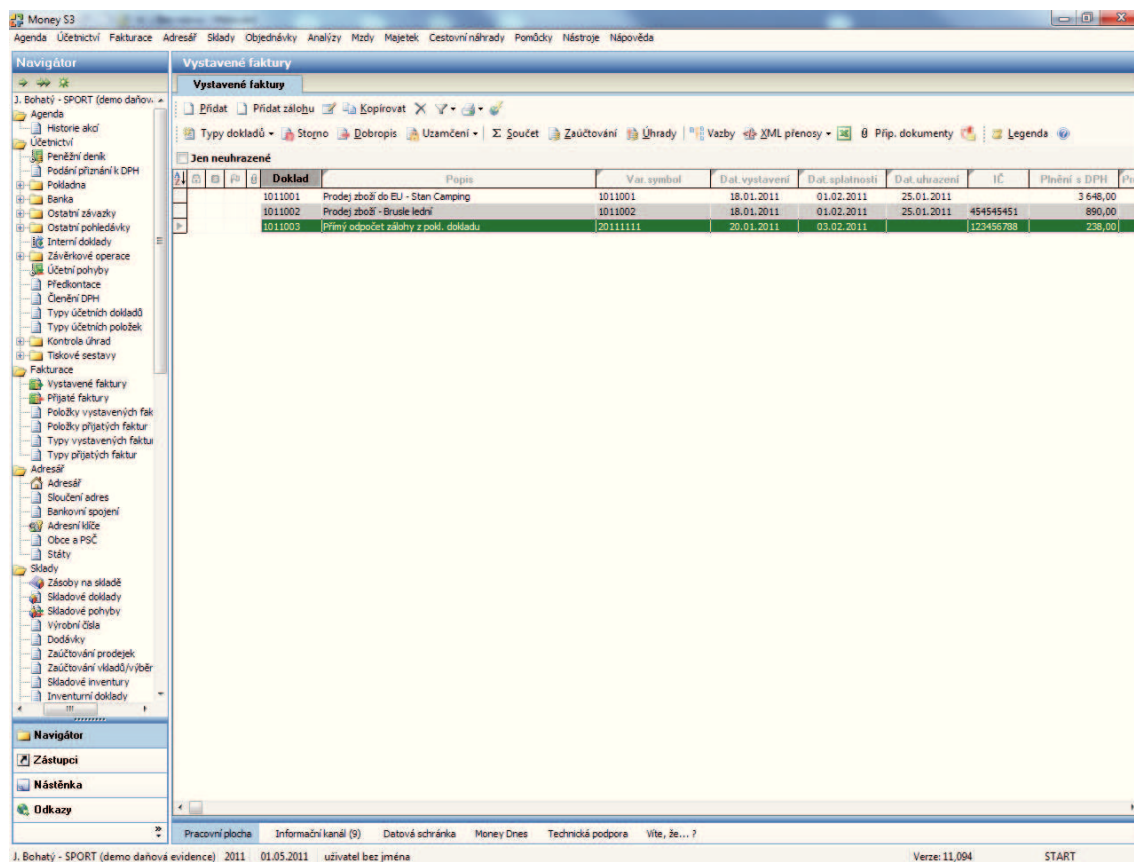
3.2 Srovnání existujících účetních programů

Na trhu existuje řada aplikací a účetních programů pro vedení účetní evidence formou DE (daňové evidence), ze kterých si může běžný uživatel nejčastěji vybrat již hotový produkt jedné z firem, které účetní SW (software) vytváří anebo si nechat účetní program vytvořit na míru. Řešení na míru je ovšem cenově nákladné a navíc jsou pro většinu firem zcela dostačující běžně dostupné produkty, které obsahují většinu potřebných evidencí. V případě, že však chce zákaznická firma například propojit informační systém s účetním programem nebo na účetní program napojit specifický internetový obchod, pak bude cena takového produktu samozřejmě výrazně vyšší.

Mezi nejznámější firmy, které o přízeň uživatelů svádějí ve výrobě účetních programů boj již několik let, patří bezesporu STORMWARE s.r.o. a CÍGLER SOFTWARE, a.s. se svými účetními programy POHODA a Money S3. Každá z těchto dvou firem má v oboru několikaleté zkušenosti, takže by se dalo s nadsázkou říci, že v tuzemsku určují tempo a směr, jakým se účetní programy vyvíjejí.

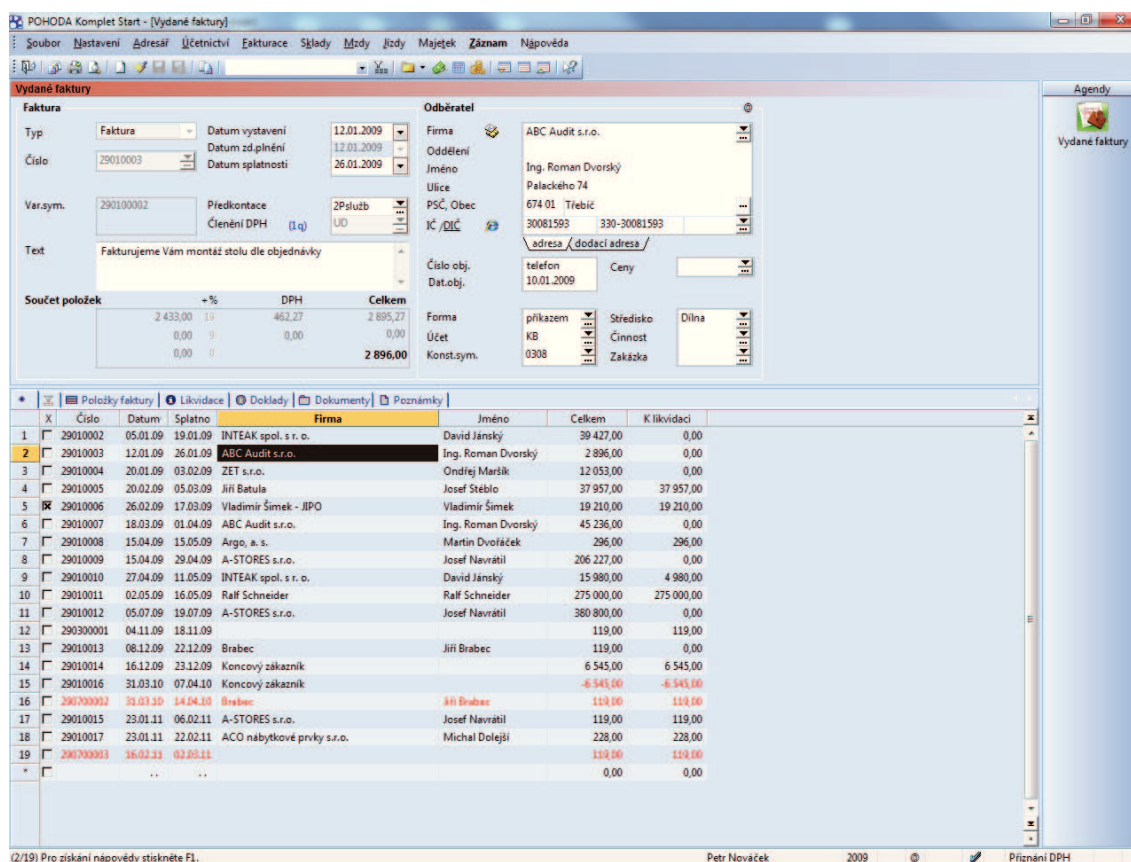
Pokusit se vytvořit účetní program, který by se s těmito produkty měl rovnat samozřejmě není nemožné, ale rozhodně se nejedná o jednoduchou záležitost, protože obě firmy již mají za sebou v tomto oboru letitou praxi a zaměstnávají týmy profesionálních analytiků, programátorů a konzultantů. Proto jsou i uživatelská rozhraní obou programů velmi přívětivá a i pro začínajícího uživatele není velký problém se v nich po krátké době zorientovat a začít pracovat. Rozhraní také prochází neustálým vývojem ve prospěch uživatele, o čemž svědčí i následující porovnání grafického uživatelského rozhraní účetních programů Money S3 START a POHODA START.

Na obrázku 1 na straně 11 je zobrazena ukázka grafického uživatelského rozhraní účetního programu Money S3 START [15]. Na této ukázce lze vidět evidenci vystavených faktur.



Obrázek 1: Money S3 START - ukázka GUI

Na obrázku 2 na straně 12 je zobrazena ukázka grafického uživatelského rozhraní účetního programu POHODA START [16]. Na této ukázce lze vidět evidenci vydaných faktur, tedy stejnou evidenci jako u předchozí ukázky účetního programu.



Obrázek 2: POHODA START - ukázka GUI

CÍGLER SOFTWARE, a.s. navíc v archivu ještě stále nabízí zdarma ke stažení verzi Money 6.603, která byla vyvíjena na počátku 90. let ještě pro operační systém MS DOS. Jednalo se o mimořádně úspěšnou verzi a někteří malí podnikatelé ji stále používají - samozřejmě s určitým omezením daným novějšími účetními zákony.

U obou porovnávaných účetních programů POHODA a Money S3 dokonce existují startovní verze, které si uživatel může zdarma vyzkoušet nebo v nich začít účtovat, ale tyto verze zpravidla mají mnoho omezení, jako například maximální počet záznamů v evidenci, apod. Verze s označením START jsou tedy velmi limitující a skutečně byly primárně navrženy jako časově neomezené demoverze, ve kterých si uživatel vyzkouší funkcionalitu a uživatelské rozhraní, aby si pak vybral placený produkt, který bude nejlépe vyhovovat jeho představám.

Stále však na trhu chybí produkt, který by byl dostupný v rámci licencí svobodného softwaru a to zejména z důvodu častých změn v legislativě účetních zákonů, které pak vyžadují velmi rychlou reakci výrobců účetního softwaru, kteří musí tyto změny v co nejkratším čase zahrnout do svých produktů a dostupných aktualizací tak, aby uživatel mohl bez časové prodlevy, ve které by se teprve vyvíjela aktualizace, stále pracovat.

Funkcionalita	POHODA Mini	Money S3 Mini
Daňová evidence	ANO	ANO
Adresář	ANO	ANO
Majetek	-	ANO
Finance	ANO	-
Daně	ANO	-
Fakturace	ANO	-
Základní cena	1980,- Kč	1990,- Kč

Tabulka 1: Srovnání funkcionality účetních programů POHODA Mini a Money S3 Mini

Vydávání aktualizací, ať už ve formě oprav chyb v účetních programech nebo po znovelizování zákonů o účetnictví, je velmi potřebnou až nedílnou částí snad všech účetních programů. Firmy, vytvářející účetní software, si jsou tohoto faktu vědomy, takže se z aktualizací stává další nemalý zdroj příjmů těchto firem.

Podrobné srovnání účetních programů by bylo velmi rozsáhlé a není ani předmětem této práce. Tabulka 1 na straně 13 proto srovnává pouze ceny nejlevnějších verzí zmiňovaných dvou produktů POHODA a Money S3, tedy jak již bylo zmíněno, jedny z nejrozšířenějších účetních programů mezi uživateli v České republice.

4 Analýza

Základem obsahu této diplomové práce je nutnost důkladně zanalyzovat zadaný problém, tedy vytvořit co nejpresnější analytický návrh pro následnou implementaci účetního informačního systému pro vedení daňové evidence. Nejprve bude nutné zjistit co nejvíce požadavků na takovýto informační systém a provést jejich specifikaci. Na základě vytvořené specifikace požadavků provést podrobnou analýzu, která přesněji stanoví strukturu a částečně i funkcionalitu IS. Poté již bude možné blíže upřesnit vytvořenou analýzu použitím vybraných metod návrhu implementace, které budou spolu s analýzou vstupem pro samotnou implementaci.

4.1 Specifikace požadavků

V průběhu několika minulých let vznikl požadavek vytvořit plně funkční informační systém daňové evidence dříve označované jako jednoduché účetnictví, který bude určen k nabízení potencionálním zákazníkům, popřípadě celý informační systém nebo jeho část poskytnut zdarma jako součást jiného produktu nebo služby. Informační systém bude podporovat uživatelská nastavení parametrů aplikace, ke kterým může dojít například po drobných změnách v účetní legislativě, tak, aniž by musel v těchto případech zasahovat do kódu programátor.

PROČ se má vytvořit informační systém pro daňovou evidenci? Podnět na vznik IS daňové evidence se objevil kvůli požadavkům potencionálních zákazníků na uživatelsky přívětivou, nenáročnou a finančně dostupnou aplikaci pro vedení účetnictví systémem daňové evidence pro malé firmy.

ČEMU má informační systém sloužit? IS daňové evidence (dříve jednoduchého účetnictví) slouží k vedení evidence účetních případů pro zjištění daně z příjmů z podnikání nebo jiné výdělečné činnosti. IS také bude sloužit pro vedení dalších - z hlediska legislativy nepovinných evidencí, jakými jsou evidence objednávek, nabídek a poptávek, zaměstnanců, pokladních a bankovních dokladů, atd.

KDO bude s informačním systémem pracovat? Pracovat s IS daňové evidence bude převážně osoba v roli účetní, ale samozřejmě je možné, aby s ním pracovali i majitelé firem, případně k tomuto úkolu zmocnění pracovníci. Vzhledem k uchovávání citlivých dat firem bude kladen velký důraz na bezpečnost celého informačního systému.

4.1.1 Vstupy aplikace

Jako vstupy do informačního systému bude sloužit mnoho seznamů počínaje seznamy určenými pouze pro nastavení detailů, ale také samozřejmě i těmi nejdůležitějšími, které mají za úkol uchovávat data o účetních případech konkrétních účetních agend. Pro ukázkou jsou zde vybrány seznamy, přímo související s fakturami. Jedná se tedy o výběr tabulek, které mají vazbu na tabulku faktura včetně této tabulky. Zbývající vstupy IS jsou uvedeny ve zvláštní příloze Specifikace požadavků, uložené na přiloženém CD (compact disc) ve složce „přílohy“ v souboru „specifikace_pozadavku.pdf“.

Seznam agenda uchovává informace o účetních jednotkách. Jako klíč zde slouží ID_agenda, který jednoznačně identifikuje účetní jednotku. Dále název (název) účetní jednotky, to může být buď jméno konkrétního člověka anebo celé firmy. Datum zřízení účetní jednotky určuje atribut datum. Poslední atribut rok uchovává aktuálně zvolený účetní rok.

Seznam agenda_firma upřesňuje parametry konkrétních účetních agend o důležité informace jakými jsou kontaktní údaje firmy a další podrobnosti důležité vždy v rozhodný okamžik vytvoření faktury, dokladu nebo dalších účetních případů. ID_agenda_firma jednoznačně identifikuje firemní identitu v rozhodný okamžik (při změně údajů je nutné provést změnu aktuální identity agendy i v tabulce nastavení, aby mohly být dále pro nové účetní případy používány nové údaje o firmě agendy). ID_agenda je cizím klíčem z tabulky agenda. Další jednotlivé atributy jsou svými názvy výstižné, takže není potřeba je podrobněji popisovat a postačí pouze komentář uvezený závorkami. Seznam atributů: název (firmy), jmeno (kontaktní osoba), ulice, obec, psc, stat (stát), ico (IČ), dic (DIČ), banka (název bankovního ústavu, kde je veden běžný podnikatelský účet), bu (číslo bankovního účtu), kod_bu (kód banky), telefon, fax, www (webové stránky), email (kontaktní email), zapis (údaje o provedení zápisu do obchodního rejstříku), aktualni (aktuální záznam o firmě: 0 - neaktuální, 1 - aktuální), logo (název souboru s logem firmy - obrázek).

Seznam dph obsahuje hodnoty daní z přidaných hodnot a jejich názvy. ID_dph jednoznačně identifikuje konkrétní vloženou daň, ID_agenda je cizím klíčem z tabulky agenda, název (název) je název uvedené daně, například „snížená daň“. Atribut sazba v procentech vyjadřuje výši sazby DPH, zakladni_snizena identifikuje, zda se jedná o základní, sníženou nebo osvobozenou sazbu DPH, jedná se o integer číslo s následující reprezentací: 1 - základní sazba DPH, 2 - snížená sazba DPH, 3 - osvobozená sazba DPH. Posledním atributem je atribut aktualni, číslo int označující 1 aktuálně používanou sazbu a 0 dříve používanou sazbu DPH.

Seznam faktura obsahuje informace o vystavených i přijatých fakturách. Každou fakturu jednoznačně identifikuje ID_faktura, ID_agenda je cizím klíčem a odkazuje se na konkrétní agendu, pod kterou byla faktura vytvořena, datum_vystaveni_prijeti označuje datum vystavení nebo přijetí faktury, která tak přichází v platnost. Atribut datum_splatnosti říká, kdy je termín splatnosti, datum_odpocet určuje, na který měsíc je možno provést odpočet daně za tuto fakturu. ID_predkontace je odkaz na důvod vytvoření faktury (zda se jedná o fakturu za zboží, materiál, apod.). Atribut popis je klasickým textovým popisem faktury a popis2 je soukromý textový popis faktury. ID_firma_USED se odkazuje na použitou firmu (ať již vybranou na základě seznamu firem, adres a kontaktů nebo zapsanou ručně). ID_forma_platby určuje, čím bude fakturu možno splatit (příkazem, hotově). Dále následují atributy variabilni_symbol (variabilní symbol), ID_konstantni_symbol, což je cizí klíč, který se odkazuje na seznam konstantních symbolů. Podle atributu prijata_vystavena se pozná, zda se jedná o vystavenou nebo přijatou fakturu (hodnoty: 1 - vystavená, 2 - přijatá), atribut ID_navazuje_faktura může nabývat hodnoty NULL a je to odkaz na ID jiné faktury (pokud se tato faktura na nějakou fakturu váže, dostane do tohoto atributu její ID_faktura). ID_sleva se odkazuje na konkrétní slevu ze seznamu slev, sleva pak platí pro celou fakturu. ID_odpocet_dph indikuje možnost odpočtu DPH k dané faktuře.

ID_objednavka může nabývat hodnoty NULL a slouží jako odkaz na konkrétní objednávku v seznamu objednávek, jestliže na objednávku faktura navazuje. ID_agenda_firma je odkazem na aktuální stav parametrů domovské firmy. Tento atribut se eviduje kvůli možným změnám v názvu nebo adrese firmy, aby byla zachována správná adresa v okamžiku vystavení faktury. Atribut cislo_faktury identifikuje fakturu v rámci posloupnosti faktur, která by neměla být porušena, číslo faktury se skládá z 10 čísel. Atribut text_uzivatel může vyplnit v případě, že chce na fakturu umístit informační text pro sebe nebo cílovou firmu. Typ faktury určuje číslem int, zda je faktura: 1 - faktura, 2 - vrubopis, 3 - dobropis, 4 - zálohová faktura, 5 - proforma faktura. Zbývající tři atributy jsou u faktury uvedeny pro korektní zaokrouhlení cen faktury. Atribut zaokrouhleni - zaokrouhlení celkové částky na x desetinných míst, atribut zaokrouhleni_typ - 1 - matematicky, 2 - nahoru, 3 - dolů. Poslední atribut polozky_zobrazeni_zakrouhleni - zobrazí x desetinných míst u položek faktury.

Seznam faktura_dph je spojovací tabulkou mezi tabulkami faktura a dph. Obsahuje dva atributy - primární klíče ID_faktura a ID_dph odkazující se na příslušné tabulky. Jedná se o rozklad vazby M:N.

Seznam faktura_mena je spojovací tabulkou mezi tabulkami faktura a mena, ale obsahuje i atributy s přidánými daty pro přesnou identifikaci typu měny (tuzemská, zahraniční, souběžná) a případným kurzem měny a množstvím kurzu měny. Primárními klíči jsou ID_faktura a ID_mena. Kurz je float číslo, které je vyplněno pouze v případě, že je faktura evidována v cizí měně. Mnozstvi_kurz je těsně spjato s kurzem a znamená množství měny vzhledem k přepočtu kurzu, například Euro má kurz 25,0 a množství kurzu 1 a slovenská Koruna před přechodem na Euro mohla mít kurz 80,0 a množství kurzu 100 oproti Koruně české. Typ_meny je typu int a může nabývat následujících hodnot: 1 - aktuální výchozí měna, 2 - cizí měna, 3 - souběžná měna.

Seznam firma_used je používán pro uložení konkrétních cílových firem v rámci faktury, dokladu, objednávky, apod., protože práce s adresářem firem nestačí k tomu, aby se v něm zohlednily změny adres, kontaktů a dalších údajů firem a například pro konkrétní fakturu je nutné, aby byl k dispozici záznam konkrétní firmy, který byl aktuální v době vytvoření faktury. ID_firma_USED jednoznačně identifikuje záznam firmy, který je použit například ve fakturách, dokladech, apod. ID_agenda je cizím klíčem a odkazuje se na konkrétní účetní agendu z tabulky agenda. Atributy nazev, jmeno, ulice, obec, psc, stat (stát), ico (IČ), dic (DIČ) mají své výstižné pojmenování, není proto dalšího komentáře. Atribut bu označuje bankovní účet a kod_bu je spjato s atributem bu a jedná se o kód banky.

Seznam forma_platby slouží k uchování forem plateb, kterými je možno platit. Jedná se například o platbu příkazem, hotově, atd. ID_forma_platby jednoznačně určuje konkrétní formu platby, atribut nazev zase popisuje tuto formu. ID_agenda je cizím klíčem a odkazuje se na konkrétní účetní agendu z tabulky agenda.

Seznam jednotka_mnozstvi slouží k shromažďování informací o množstevních jednotkách používaných v rámci účetnictví. Atribut ID_jednotka_mnozstvi jednoznačně identifikuje jednotku množství a je zároveň klíčem. Atribut nazev popisuje danou jednotku,

například "počet kusů", zatímco atribut znacka je zkratkou ke svému názvu, například „ks“.

Kontanтни_symbol je seznam sloužící zejména pro čtení konstantních symbolů, který je důležitý zejména při transakcích, proto je používán například u faktur. Jednoznačně identifikuje každý konstantní symbol ID_kontanтни_symbol, ID_agenda je pak cizím klíčem a odkazuje se na konkrétní účetní agendu z tabulky agenda. Atribut nazev je používán pro přehlednou evidenci konstantních symbolů a vysvětluje význam konkrétního čísla atributu konstantni_symbol, který už je pouhým označovacím číslem symbolu.

V seznamu mena jsou uloženy všechny používané měny. ID_mena jednoznačně identifikuje konkrétní měnu a ID_agenda je cizím klíčem z tabulky agenda. Atributy zkratka, nazev a oznaceni slouží k označení měny, ať už názvem (Koruna česká), zkratkou (Kč) nebo označením (CZK). Pokud je měna souběžná, může využívat atributy pevny_kurz, mnozstvi (množství kurzu), které v opačném případě nabývají hodnot NULL. Atribut aktualni je typu int a označuje, zda je měna aktuální výchozí měnou. Tento atribut může nabývat hodnot: 1 - ano (výchozí měna), 2 - ne (cizí měna), 3 - souběžná měna.

Seznam objednavka obsahuje informace o vystavených i přijatých objednávkách. Každou objednávku jednoznačně identifikuje ID_objednavka, ID_agenda je cizím klíčem a odkazuje se na konkrétní agendu, pod kterou byla objednávka vytvořena, datum_vytvoreni označuje datum vytvoření nebo přijetí objednávky. Atribut datum_platnosti říká, do kdy je objednávka platná. Atribut popis je klasickým textovým popisem objednávky a popis2 je soukromý textový popis objednávky. ID_firma_USED se odkazuje na použitou firmu (ať již vybranou na základě seznamu firem, adres a kontaktů nebo zapsanou ručně). Podle atributu prijata_vystavena lze poznat, zda se jedná o vystavenou nebo přijatou objednávku (hodnoty: 1 - vystavená, 2 - přijatá). ID_dph_sada je odkazem na tabulku dph_sada, ve které jsou informace o objednávkou tehdy používaných sazbách DPH. ID_agenda_firma je odkazem na aktuální stav parametrů domovské firmy. Tento atribut se eviduje kvůli možným změnám v názvu nebo adrese firmy, aby byla zachována správná adresa v okamžiku vystavení nebo přijetí objednávky. Atribut cislo_objednavky identifikuje objednávku v rámci posloupnosti objednávek, která by neměla být porušena a číslo objednávky se skládá z 10 čísel. Atribut text_uzivatel může vyplnit v případě, že chce objednávku označit informačním textem (malá poznámka). Zbývající tři atributy jsou u objednávky uvedeny pro korektní zaokrouhlení cen objednávky. Atribut zaokrouhleni - zaokrouhlení celkové částky na x desetinných míst, atribut zaokrouhleni_typ - 1 - matematicky, 2 - nahoru, 3 - dolů. Poslední atribut polozky_zobrazeni_zaokrouhleni - zobrazí x desetinných míst u položek objednávky.

Odpocet_dph je jednoduchým seznamem pro označení jednotlivých účetních případů v rámci odpočtu DPH. Zda-li se tedy bude zahrnovat do přiznání DPH nebo ne, zda-li se jedná o tuzemské nebo zahraniční plnění, apod. Využívá pouze 3 atributy. Jednoznačně identifikovaný primární klíč ID_odpocet_dph, ID_agenda - cizí klíč z tabulky agenda a nazev odpočtu DPH.

Položky bankovního dokladu jsou reprezentovány seznamem polozka_bankovni_doklad. Jednoznačně každou položku identifikuje primární klíč ID_polozka_bankovni_doklad, ID_bankovni_doklad je cizím klíčem z tabulky bankovni_doklad, který přiřazuje

jednotlivé položky ke konkrétnímu bankovnímu dokladu. Atribut `ID_jednotka_mnozstvi` jednoznačně identifikuje jednotku množství. `ID_dph` odkazuje na konkrétní sazbu DPH v seznamu `dph`, která se k položce vztahuje. `ID_odpocet_dph` se odkazuje na seznam `odpocet_dph` a u každé položky dokladu je tak určeno, zda-li budou zahrnuty do přiznání DPH nebo nikoliv. Na základě uživatelem vybraného atributu `ID_predkontace` jsou pak jednotlivé položky dokladu zobrazeny v konkrétních kolonkách peněžního deníku. `ID_agenda` je cizím klíčem a odkazuje se na konkrétní agendu, pod kterou byl doklad a každá jeho konkrétní položka vytvořena. Atribut `nazev` pojmenovává položku bankovního dokladu a atribut `text` ji může dodatečně popisovat. Množství se udává v celých nebo desetinných číslech a jednotkou pro množství je vybraný atribut `ID_jednotka_mnozstvi`. Atribut `zakladni_cena_ks` je základní cenou bez DPH. Sleva, pokud je určena a není 0, je uvedena v % a může být celým nebo desetinným číslem. Jedná se o slevu z ceny položky bankovního dokladu.

Seznam vazeb položek bankovních dokladů na konkrétní faktury je nazván `polozka_bd_likvidace_faktura` a je spojovací tabulkou mezi tabulkami `polozka_bankovni_doklad` a `faktura`. Má dva primární klíče `ID_polozka_bankovni_doklad` a `ID_faktura` odkazující se na příslušné tabulky.

Položky faktury jsou reprezentovány seznamem `polozka_faktury`. Jednoznačně každou položku identifikuje primární klíč `ID_polozka_faktury`, `ID_faktura` je cizím klíčem z tabulky `faktura`, který přiřazuje jednotlivé položky ke konkrétní faktuře. Atribut `ID_jednotka_mnozstvi` jednoznačně identifikuje jednotku množství. `ID_dph` odkazuje na konkrétní sazbu DPH v seznamu `dph`, která se k položce vztahuje. `ID_odpocet_dph` se odkazuje na seznam `odpocet_dph` a u každé položky faktury je tak určeno, zda-li budou zahrnuty do přiznání DPH nebo nikoliv. Na základě uživatelem vybraného atributu `ID_predkontace` jsou pak jednotlivé položky faktury zobrazeny v konkrétních kolonkách peněžního deníku. `ID_zaruka_typ` označuje typ záruky (v počtech měsíců, let, doživotní, apod.). `ID_agenda` je cizím klíčem a odkazuje se na konkrétní agendu, pod kterou byla faktura a každá její konkrétní položka vytvořena. Atribut `nazev` pojmenovává položku faktury a atribut `text` ji může dodatečně popisovat. Množství se udává v celých nebo desetinných číslech a jednotkou pro množství je vybraný atribut `ID_jednotka_mnozstvi`. Atribut `zakladni_cena_ks` je základní cenou bez DPH. Sleva, pokud je určena a není 0, je uvedena v % a může být celým nebo desetinným číslem. Jedná se o slevu z ceny položky faktury. Posledním atributem je `zaruka`, která jak název napovídá určuje v celých číslech záruku. Typ záruky určuje již zmiňovaný atribut `ID_zaruka_typ`, který upřesňuje, zda je záruka uvedena v měsících, letech nebo je doživotní.

Seznam vazeb položek pokladních dokladů na konkrétní faktury je nazván `polozka_pd_likvidace_faktura` a je spojovací tabulkou mezi tabulkami `polozka_pokladni_doklad` a `faktura`. Má dva primární klíče `ID_polozka_pokladni_doklad` a `ID_faktura` odkazující se na příslušné tabulky.

Položky pokladního dokladu jsou reprezentovány seznamem `polozka_pokladni_doklad`. Jednoznačně každou položku identifikuje primární klíč `ID_polozka_pokladni_doklad`, `ID_pokladni_doklad` je cizím klíčem z tabulky `pokladni_doklad`, který přiřazuje jednotlivé položky ke konkrétnímu pokladnímu dokladu. Atribut `ID_jednotka_mnozstvi` jednozna-

čně identifikuje jednotku množství. ID_dph odkazuje na konkrétní sazbu DPH v seznamu dph, která se k položce dokladu vztahuje. ID_odpocet_dph se odkazuje na seznam odpocet_dph a u každé položky dokladu je tak určeno, zda-li budou zahrnuty do přiznání DPH nebo nikoliv. Na základě uživatelem vybraného atributu ID_predkontace jsou pak jednotlivé položky dokladu zobrazeny v konkrétních kolonkách peněžního deníku. ID_agenda je cizím klíčem a odkazuje se na konkrétní agendu, pod kterou byl doklad a každá jeho konkrétní položka vytvořena. Atribut nazev pojmenovává položku pokladního dokladu a atribut text ji může dodatečně popisovat. Množství se udává v celých nebo desetinných číslech a jednotkou pro množství je vybraný atribut ID_jednotka_mnozstvi. Atribut zakladni_cena_ks je základní cenou bez DPH. Sleva, pokud je určena a není 0, je uvedena v % a může být celým nebo desetinným číslem. Jedná se o slevu z ceny položky pokladního dokladu.

Seznam predkontace je využíván účetními případy, které se promítnou v peněžním deníku. Na základě vybrané předkontace jsou účetní případy zobrazeny v konkrétních kolonkách peněžního deníku, například zboží, materiál, režie, apod. Každou předkontaci jednoznačně identifikuje ID_predkontace, ID_agenda je pak cizím klíčem a odkazuje se na konkrétní účetní agendu z tabulky agenda. Typ je atribut určující příjem nebo výdaj. Atributy zkratka a nazev již pro uživatele srozumitelně označují vybranou předkontaci.

Seznam zaruka_typ slouží pro pojmenování typů záruk k délce záruky a obsahuje v atributu nazev hodnoty jako například "měsíců", "let", "doživotní". Jednoznačným primárním klíčem je ID_zaruka_typ a cizím klíčem pro ověření účetní agendy je ID_agenda z tabulky agenda.

4.1.2 Výstupy aplikace

Jako výstupy nám budou sloužit různé výpisy seznamů, formulářů, informační okna a tiskové sestavy, se kterými budou uživatelé dále pracovat.

- peněžní deník (výpisy všech provedených účetních případů, které ovlivňují výši daně z příjmů)
- správa proměnlivých položek uživatelem - nastavení
- výpisy nastavení parametrů pro jednotlivé evidence
- výpisy a tiskové sestavy faktur
- výpisy objednávek a nabídek - přijaté, vystavené
- výpisy bankovních a pokladních dokladů - přijatých, vystavených
- výpisy zaměstnanců - seznamy zaměstnanců, mzdy
- výpisy vozidel - vozidla, kniha jízd
- výpisy majetku - hmotný, nehmotný, leasingový

- adresář - možnost vedení kontaktů a jejich aktivní používání - výpisy do formulářů, výpisy seznamů
- kalendář pro výběr požadovaného data a možností výběru do formuláře
- výpisy stavů skladu

4.1.3 Funkce

Informační systém pro daňovou evidenci musí obsahovat základní funkce pro korektní a komfortní práci uživatele se systémem tak, aby splnil účel, pro který byl IS vytvořen. Součástí nového informačního systému však budou i nadstandardní funkce, které mají za úkol zvýšit úroveň a kvalitu informačního systému tak, aby co nejvíce vyhovoval uživatelům, tedy osobám, které vedou účetnictví formou daňové evidence. Z pohledu informačního systému je jediným možným aktérem, pracujícím se systémem a využívajícím jeho funkce účetní jednotka. Účetní jednotka může využívat následující funkce¹:

- přihlášení
- pokladní doklad - přidání, editace, smazání, výpis a vyhledání pokladního dokladu
- položka pokladního dokladu - přidání, editace, smazání a výpis položek pokladního dokladu, likvidace faktury položkou pokladního dokladu
- bankovní doklad - přidání, editace, smazání, výpis a vyhledání bankovního dokladu
- položka bankovního dokladu - přidání, editace, smazání a výpis položek bankovního dokladu, likvidace faktury položkou bankovního dokladu
- přijatá faktura - přidání, editace, smazání a výpis přijatých faktur, stornování faktury
- vystavená faktura - přidání, editace, smazání a výpis vystavených faktur, stornování faktury, export faktury do PDF
- položka faktury - přidání, editace, smazání a výpis položek faktury
- objednávka - přidání, editace, smazání, výpis a vyhledání objednávky
- položka objednávky - přidání, editace, smazání a výpis položek objednávky
- nabídka a poptávka - přidání, editace, smazání, výpis a vyhledání nabídky nebo poptávky
- položka nabídky/poptávky - přidání, editace, smazání a výpis položek nabídky nebo poptávky

¹Pro zjednodušení jsou jednotlivé funkce přiřazeny ke svým typům entit.

- adresář - přidání, editace, smazání, výpis a vyhledání firem; přidání, editace, smazání, výpis a vyhledání adres konkrétní firmy; přidání, editace, smazání, výpis a vyhledání kontaktu, přiřazení kontaktu k jiné firmě
- zaměstnanec - přidání, editace, smazání, výpis a vyhledání zaměstnanců
- mzda zaměstnance - přidání, editace, smazání, výpis a výpočet mezd zaměstnance, odvody z mezd zaměstnanců
- kniha jízd - přidání, editace, smazání, výpis a vyhledání jízd, řidičů a vozidel
- majetek - přidání, editace, smazání, výpis, vyhledání majetku, odpisy majetku
- sklad - přidání, editace, smazání, výpis a vyhledání materiálu/zboží na skladě, příjemky, výdejky
- peněžní deník - výpis a vyhledání položek v peněžním deníku
- nastavení pokladen - přidání, editace, smazání, výpis a zneplatnění pokladen
- nastavení bankovních účtů - přidání, editace, smazání, výpis a zneplatnění bankovních účtů
- nastavení používaných měn
- zálohování účetnictví
- roční uzávěrky - uzavření účetního roku, započetí nového účetního roku
- obecná nastavení - editace nastavení základních parametrů (zaokrouhlování cen, evidenčních čísel, apod.); přidání, editace, smazání a výpis základních seznamů s nastavením
- odhlášení

4.1.4 Okolí systému

Vztah a komunikaci informačního systému s okolím znázorňuje kontextový diagram. Kontextový diagram je součástí funkční analýzy, obrázek 4 na straně 30. V okolí informačního systému existuje pouze jeden aktér s názvem účetní jednotka. Pro přiblížení - účetní jednotka je uživatel přihlášený do informačního systému daňové evidence.

4.1.5 Bezpečnost aplikace

Každý uživatel, který chce pracovat s informačním systémem daňové evidence musí být přihlášen, aby na základě ověření svých přihlašovacích údajů (přihlašovacího jména a hesla) mohl začít s IS pracovat. Po přihlášení bude uživateli přiřazeno jednoznačné ID, pomocí kterého bude autorizován pro práci s jednotlivými prvky systému. IS, tedy jeho vstupy budou zabezpečeny proti SQL injectům a XSS útokům. Jedná se zejména o ošetření

metod GET, POST a formulářových prvků, které tyto metody využívají pro uchování dat po odeslání formuláře.

4.1.6 Nefunkční požadavky

Vzhledem k absenci IS daňové evidence, umožňující nenákladné budoucí rozšíření na více stanic s možností použití webového serveru v místní síti vznikl požadavek na vytvoření informačního systému, který by poskytoval do budoucna možnost rozšíření na několik počítačů bez nutnosti instalace účetního softwaru, spojené s dalšími náklady a přikupováním licencí. Celý informační systém bude naimplementován v skriptovacím programovacím jazyce PHP verze 5 za použití značkovacího jazyka XHTML[9]. Jako systém pro řízení báze dat bude použita MySQL. V rámci implementace se budou využívat i další moderní technologie jakými jsou například knihovna jQuery[13], AJAX skripty, šablonovací systém Smarty[10], apod. Jako styl bude použit kaskádový styl CSS[11].

Informační systém bude optimalizován pro rozlišení 1280x1024 a 1024x768, které jsou v současné době mezi běžnými uživateli stále hojně používány, i když dnešním trendem je přechod na širokoúhlý obraz.

Mezi jednu z největších výhod webových aplikací a informačních systémů patří bezesporu jejich dostupnost a nezávislost na platformě (OS). Bude zajištěna kompatibilita s prohlížeči Internet Explorer 7, Internet Explorer 8, Opera, Mozilla Firefox, Google Chrome.

Vnější požadavkem je časový termín, stanovený na termín odevzdání této práce, tedy 6. května 2011.

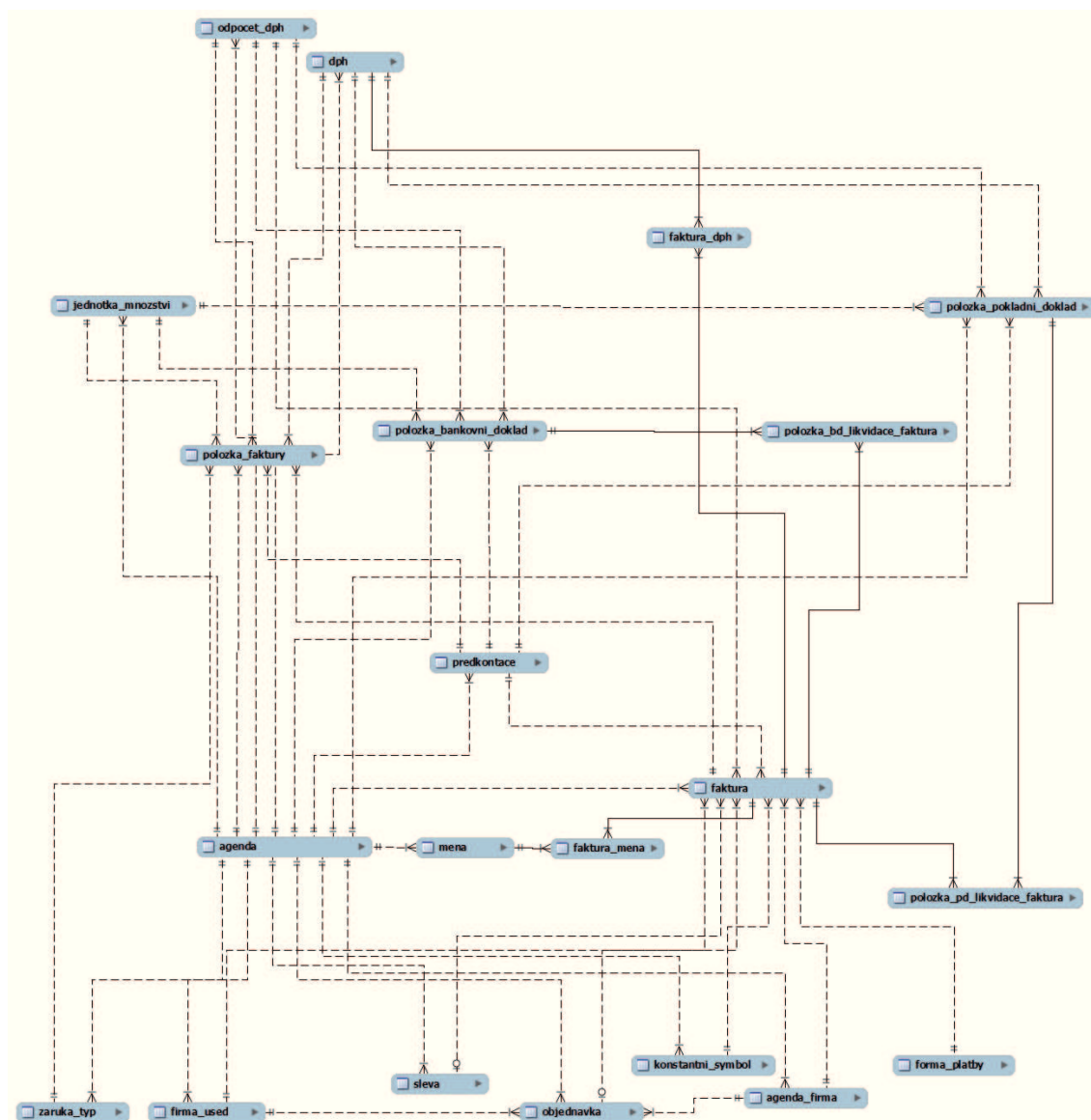
4.2 Datová analýza

V této fázi analýzy bude navržena základní podoba databáze, tedy základní podoba typů entit. Pro tento účel běžně postačuje ER-diagram, datový slovník a lineární zápis typů entit. Vzhledem k rozsáhlosti ER diagramu a datového slovníku (44 typů entit) bude opět uvedena pouze zkrácená část, zahrnující tabulky přímo související s fakturami a k faktuře příslušným funkcím. Kompletní datová analýza je součástí přílohy na CD.

4.2.1 ER diagram - evidence faktur

Obrázek 3, který je uveden níže, znázorňuje ukázkou části ER diagramu. Vzhledem k rozsáhlosti a zachování přehlednosti ER diagramu není vhodné jej zde celý zobrazovat. Na CD jsou ve složce „přílohy“ soubory, ve kterých je k dispozici celý ER-diagram. Jedná se o obrázek typu PNG v souboru „er_diagram.png“ a dále soubor typu „er_diagram.mwb“. Se souborem typu MWB dokáže pracovat nástroj MySQL Workbench[4] určený pro práci s databázemi typu MySQL. Jedná se o soubor s vytvořeným projektem, ER-diagramem a kompletní databází. ER-diagram se tak stane pomocí interaktivního zvýraznění syntaxe přehlednější a orientace v celém ER-diagramu je pak jednodušší, nehledě na to, že lze i nahlédnout do seznamu atributů jednotlivých tabulek a jejich závislostí. Vzhledem k lepší orientaci a přehlednosti ER-diagramu je tedy vhodnější použít soubor MWB.

Na obrázku 3 na straně 23 je zobrazena část ER-diagramu, zaměřující se pouze na evidenci faktur.



Obrázek 3: Ukázka části ER diagramu - evidence faktur

4.2.2 Lineární zápis typů entit

Lineární zápis typů entit reprezentuje jednotlivé tabulky databáze s jejich atributy. Pro přehlednost zápisů jsou primární klíče zvýrazněny kurzívou.

adresa_firmy (*ID_adresa, ID_firma, ID_agenda, nazev, poradi_adresy, ulice, obec, psc, kraj, stat, telefon, fax, mobil, email, www*)

agenda (*ID_agenda, nazev, heslo, datum, rok*)

agenda_firma (*ID_agenda_firma, ID_agenda, nazev, jmeno, ulice, obec, psc, stat, ico, dic, banka, bu, kod_bu, telefon, fax, www, email, zavis, aktualni, logo*)

bankovni_doklad (*ID_bankovni_doklad, ID_bankovni_ucet, ID_konstantni_symbol, ID_predkontace, ID_odpocet_dph, ID_firma_USED, ID_dph_sada, ID_agenda, evidencni_cislo_dokladu, cislo_vypis, cislo_vypis_polozka, variabilni_symbol, specificky_symbol, text, datum_vystaveni, datum_platby, datum_odpocet, prijaty_vystaveny, popis, popis2, zaokrouhleni, zaokrouhleni_typ, polozky_zobrazeni_zaokrouhleni*)

bankovni_doklad_mena (*ID_bankovni_doklad, ID_mena, kurz, mnozstvi, typ_meny*)

bankovni_ucet (*ID_bankovni_ucet, ID_mena, ID_agenda, nazev, zkratka, evidencni_cislo_dokladu_aktualni, bu, kod_bu, iban, swift, pevny_kurz, mnozstvi, datum_zruseni*)

dph (*ID_dph, ID_agenda, nazev, sazba, zakladni_snizena, aktualni*)

dph_sada (*ID_dph_sada, ID_dph*)

faktura (*ID_faktura, ID_agenda, ID_firma_USED, ID_predkontace, ID_forma_platby, ID_konstantni_symbol, ID_sleva, ID_odpocet_dph, ID_objednavka, ID_agenda_firma, ID_navazuje_faktura, cislo_faktury, datum_vystaveni_prijeti, datum_splatnosti, datum_odpocet, popis, popis2, text, variabilni_symbol, prijata_vystavena, typ, zaokrouhleni, zaokrouhleni_typ, polozky_zobrazeni_zaokrouhleni*)

faktura_dph (*ID_faktura, ID_dph*)

faktura_mena (*ID_faktura, ID_mena, kurz, mnozstvi, typ_meny*)

firma (*ID_firma, ID_agenda, nazev, ico, dic, popis, popis2, zamereni, bu, kod_bu*)

firma_used (*ID_firma_USED, ID_agenda, nazev, jmeno, ulice, obec, psc, stat, ico, dic, bu, kod_bu*)

forma_platby (*ID_forma_platby, ID_agenda, nazev*)

jednotka_mnozstvi (*ID_jednotka_mnozstvi, ID_agenda, nazev, znacka*)

jizda (*ID_jizda, ID_agenda, datum_od, datum_do, cas_od, cas_do, ID_vozidlo, ID_ridic, trasa, cil, pocet_km, soukrome, prives, palivo_cerpano, palivo_misto, palivo_cena*)

konstantni_symbol (*ID_konstantni_symbol, ID_agenda, nazev, konstantni_symbol*)

kontakty (*ID_kontakt, ID_firma, ID_agenda, jmeno, prijmeni, titul, telefon, mobil, email, poznamka, funkce, ulice, obec, psc*)

mena (*ID_mena, ID_agenda, zkratka, nazev, oznaceni, pevny_kurz, mnozstvi, aktualni*)

nabidka_poptavka (*ID_nabidka_poptavka, ID_agenda, ID_firma_USED, ID_agenda_firma, ID_dph_sada, cislo_nabidky_poptavky, datum_vytvoreni, datum_platnosti, prijata_vystavena, popis, popis2, text, zaokrouhleni, zaokrouhleni_typ, polozky_zobrazeni_zaokrouhleni*)

nabidka_poptavka_mena (*ID_nabidka_poptavka, ID_mena, kurz, mnozstvi, typ_meny*)

nastaveni (*ID_nastaveni, ID_agenda, ID_jednotka_mnozstvi, ID_dph_sada, faktura_vystavena_faktura_cislo, faktura_vystavena_vrubopis_cislo, faktura_vystavena_dobropis_cislo, faktura_vystavena_faktura_text, faktura_vystavena_vrubopis_text, faktura_vystavena_dobropis_text, faktura_vystavena_zaokrouhleni, faktura_vystavena_zaokrouhleni_typ, faktura_vystavena_polozky_zobrazeni_zaokrouhleni, faktura_prijata_faktura_cislo, faktura_prijata_vrubopis_cislo, faktura_prijata_dobropis_cislo, faktura_prijata_faktura_text, faktura_prijata_vrubopis_text, faktura_prijata_dobropis_text, faktura_prijata_zaokrouhleni, faktura_prijata_zaokrouhleni_typ, faktura_prijata_polozky_zobrazeni_zaokrouhleni, pokladni_doklad_zaokrouhleni, pokladni_doklad_zaokrouhleni_typ, pokladni_doklad_polozky_zobrazeni_zaokrouhleni, bankovni_doklad_zaokrouhleni, bankovni_doklad_zaokrouhleni_typ, bankovni_doklad_polozky_zobrazeni_zaokrouhleni, objednavka_vystavena_cislo, objednavka_vystavena_text, objednavka_prijata_cislo, objednavka_prijata_text, objednavka_zaokrouhleni, objednavka_zaokrouhleni_typ, objednavka_polozky_zobrazeni_zaokrouhleni, nabidka_cislo, nabidka_text, poptavka_cislo, poptavka_text, nabidka_poptavka_zaokrouhleni, nabidka_poptavka_zaokrouhleni_typ, nabidka_poptavka_polozky_zobrazeni_zaokrouhleni, faktura_vystavena_zaruka, faktura_prijata_zaruka, zamestnanec_evidencni_cislo, zamestnanec_mzda_zaokrouhleni, zamestnanec_mzda_zaokrouhleni_typ, zamestnanec_socialni_pojistovna, zamestnanec_odvod_dane*)

objednavka (*ID_objednavka, ID_agenda, ID_firma_USED, ID_agenda_firma, ID_dph_sada, cislo_objednavky, datum_vytvoreni, datum_platnosti, prijata_vystavena, popis, popis2, text, zaokrouhleni, zaokrouhleni_typ, polozky_zobrazeni_zaokrouhleni*)

objednavka_mena (*ID_objednavka, ID_mena, kurz, mnozstvi, typ_meny*)

odpocet_dph (*ID_odpocet_dph, ID_agenda, nazev*)

pokladna (*ID_pokladna, ID_mena, ID_agenda, nazev, zkratka, evidencni_cislo_dokladu_aktualni, pevny_kurz, mnozstvi, datum_zruseni*)

pokladni_doklad (*ID_pokladni_doklad, ID_pokladna, ID_predkontace, ID_odpocet_dph, ID_firma_USED, ID_dph_sada, ID_agenda, evidencni_cislo_dokladu, text, datum_vystaveni, datum_platby, datum_odpocet, prijaty_vystaveny, popis, popis2, zaokrouhleni, zaokrouhleni_typ, polozky_zobrazeni_zaokrouhleni*)

pokladni_doklad_mena (*ID_pokladni_doklad, ID_mena, kurz, mnozstvi, typ_meny*)

polozka_bankovni_doklad (*ID_polozka_bankovni_doklad*, ID_bankovni_doklad, ID_jednotka_mnozstvi, ID_dph, ID_odpocet_dph, ID_predkontace, ID_agenda, nazev, text, mnozstvi, zakladni_cena_ks, sleva)

polozka_bd_likvidace_faktura (*ID_polozka_bankovni_doklad*, *ID_faktura*)

polozka_faktury (*ID_polozka_faktury*, ID_faktura, ID_jednotka_mnozstvi, ID_dph, ID_odpocet_dph, ID_predkontace, ID_zaruka_typ, ID_agenda, nazev, text, mnozstvi, zakladni_cena_ks, sleva, zaruka)

polozka_nabidka_poptavka (*ID_polozka_nabidka_poptavka*, ID_nabidka_poptavka, ID_jednotka_mnozstvi, ID_dph, ID_zaruka_typ, ID_agenda, nazev, text, mnozstvi, zakladni_cena_ks, sleva, zaruka)

polozka_objednavka (*ID_polozka_objednavka*, ID_objednavka, ID_jednotka_mnozstvi, ID_dph, ID_zaruka_typ, ID_agenda, nazev, text, mnozstvi, zakladni_cena_ks, sleva, zaruka)

polozka_pd_likvidace_faktura (*ID_polozka_pokladni_doklad*, *ID_faktura*)

polozka_pokladni_doklad (*ID_polozka_pokladni_doklad*, ID_pokladni_doklad, ID_jednotka_mnozstvi, ID_dph, ID_odpocet_dph, ID_predkontace, ID_agenda, nazev, text, mnozstvi, zakladni_cena_ks, sleva)

predkontace (*ID_predkontace*, ID_agenda, typ, zkratka, nazev)

ridic (*ID_ridic*, ID_agenda, jmeno)

sleva (*ID_sleva*, ID_agenda, nazev, sleva)

vozidlo (*ID_vozidlo*, ID_agenda, spz, palivo, spotreba, objem, poc_km, vlastnik, prives, prives_navic, typ)

zamestnanec (*ID_zamestnanec*, ID_agenda, ID_zamestnanec_pojistovna, jmeno, prijmeni, evidencni_cislo, funkce, ulice, obec, psc, bu, kod_bu, telefon, cislo_op, rodne_cislo, datum_narozeni, narodnost, statni_obcanstvi, pohlavi, rocni_zuctovani_zaloh, druh_prace, datum_zac_pomer, datum_kon_pomer, uvazek_hodina, uvazek_den, mzda, druh_mzda, uhrada_zaloha, uhrada_doplatku, vyse_zaloha, stara_dovolena, nova_dovolena, premie, pocet_deti, pocet_deti_ztp, invalidita, prukaz_ztp, student)

zamestnanec_mzda (*ID_zamestnanec_mzda*, ID_zamestnanec, ID_agenda, hruba_mzda, premie, mesic_rok, pocet_deti, pocet_deti_ztp, invalidita, prukaz_ztp, student, auto, cena_auto, dovolena, delka_pracovni_doby, pocet_pracovnich_dnu, pocet_dnu_dovolene, pocet_odpracovanych_hodin, nemocenska, pocet_hodin_nemocenska, socialni_pojisteni, socialni_pojisteni_zamestnavatel, zdravotni_pojisteni, zdravotni_pojisteni_zamestnavatel, zaloha_na_dan, sleva_dan, dan_zvyhodneni, dan_bonus, superhruba_mzda, cista_mzda_bez bonusu, cista_mzda, nahrada_mzda, castka_vyplatit, mzdo ve_naklady_zamestnavatel, socialni_pojisteni_sazba, socialni_pojisteni_zamestnavatel_sazba, zdravotni_pojisteni_sazba, zdravotni_pojisteni_zamestnavatel_sazba)

zaruka_typ (*ID_zaruka_typ*, ID_agenda, nazev)

Vlastnosti atributů jsou charakterizovány v datovém slovníku. U každého atributu je tak uveden jeho typ, velikost (délka), informace, zda je atribut klíč, zda-li je atribut indexován, jestli může nabývat hodnoty NULL a případně může být zapsána i jeho výchozí hodnota. Atribut také může obsahovat komentář s doplňujícími informacemi a případnými omezeními či stanovením formátu zobrazení. Následující datový slovník, který je zobrazen v tabulce 2 na straně 29, je vyjmut z kompletního datového slovníku, dostupném na přiloženém CD a jedná se pouze o ukázkou části datového slovníku, týkající se evidence faktur.

Sloupec	Typ	Klíč	Index	NULL	Vých.	Komentáře
faktura						
ID_faktura	int(10)	A	A	N	-	PK, AI
ID_agenda	int(8)	N	A	N	-	CK z agenda
ID_firma_USED	int(10)	N	A	N	-	CK z firma_used
ID_predkontace	int(10)	N	A	N	-	CK z predkon- tace
ID_forma_platby	int(10)	N	A	N	-	CK z forma_platby
ID_konstantni	int(10)	N	A	N	-	CK z kon- stantni_symbol
ID_sleva	int(10)	N	A	A	NULL	CK z sleva
ID_odpocet_dph	int(10)	N	A	N	-	CK z odpo- cet_dph
ID_objednavka	int(10)	N	A	A	NULL	CK z objed- navka
ID_agenda_firma	int(10)	N	A	N	-	CK z agenda_firma
ID_navazuje_fak	int(10)	N	A	A	NULL	CK z faktura; navazující faktura (např.: storno fak- tura)

Pokračování na další stránce

Pokračování na další stránce

Sloupec	Typ	Klíč	Index	NULL	Vých.	Komentáře
mnozstvi	int(10)	N	N	A	NULL	množství pro přepočet kurzu
typ_meny	int(1)	N	N	N	-	1 - aktuální výchozí měna, 2 - cizí měna, 3 - souběžná měna

Tabulka 2: Datový slovník

Vysvětlivky:

PK - primární klíč, CK - cizí klíč, AI - auto increment, n - číslice

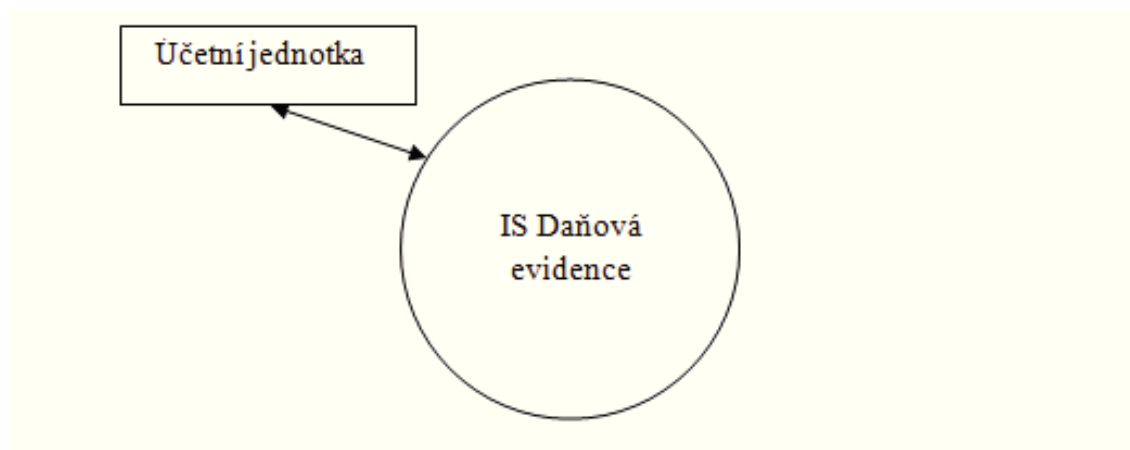
4.3 Funkční analýza

Za pomoci nástrojů funkční analýzy [6] bude provedeno upřesnění zadaných požadavků. Je zapotřebí popsat do detailu jednotlivé funkce systému tak, aby bylo možné přejít dále k návrhu implementace a později k samotné implementaci právě na základě funkční analýzy. Ve funkční analýze budou využity diagramy datových toků (DFD), popisy algoritmů jednotlivých funkcí (minispecifikace) a také časová analýza, tvořená pomocí stavových diagramů (STD).

4.3.1 Kontextový diagram

V kontextovém diagramu je znázorněn informační systém jako proces se zdrojem a cílem. Jediným aktérem, který bude s informačním systémem pracovat je uživatel v roli účetní jednotky. Konkrétní účetní jednotka je zprostředkovatelem dané firmy, které zpracovává účetnictví, s finančním úřadem a informační systém je jejím nástrojem pro korektní vstupy dat a výstupy informací a sestav z IS.

Na obrázku 4 na straně 30 je kontextový diagram, který znázorňuje informační systém se svým okolím.



Obrázek 4: Kontextový diagram

4.3.2 Diagram datových toků (DFD)

Grafické znázornění procesů, aktérů, datových úložišť a toků dat mezi nimi zprostředkovávají následující DFD diagramy. Informační systém v sobě zahrnuje celou řadu evidencí, které jsou rozepsány do DFD, roztríděných podle úrovní pohledu.

Základní popis jednotlivých evidencí:

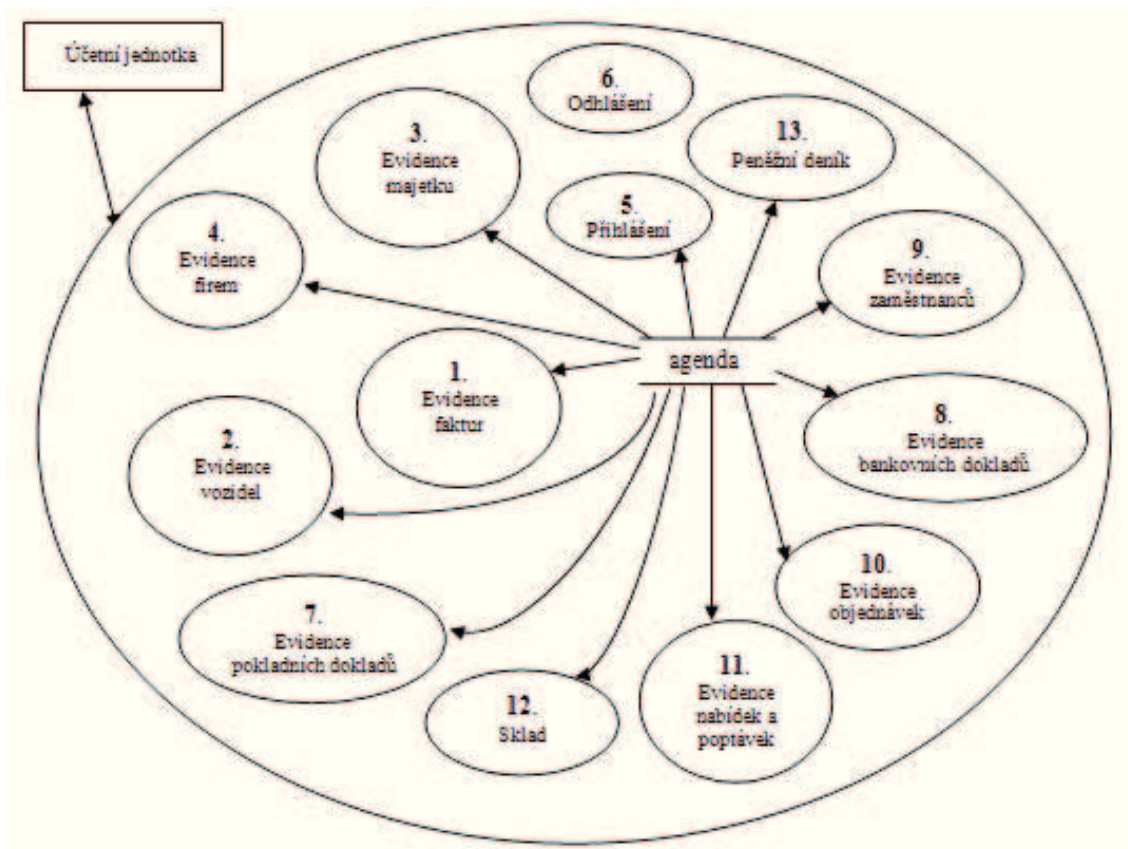
Před uvedením DFD diagramu jsou pro snazší představu vyjmenovány hlavní funkce jednotlivých procesů:

1. Evidence faktur - evidence faktur, jejich položek a likvidací
2. Evidence vozidel - evidence vozidel, jejich stavu, jízd a řidičů
3. Evidence majetku - evidence investičního majetku a jeho odpisy (hmotný, nehmotný, drobný)
4. Evidence firem - evidence firem, jejich popisu, adres a kontaktů
5. Přihlášení - přihlášení uživatele
6. Odhlášení - odhlášení uživatele
7. Evidence pokladních dokladů - evidence pokladních dokladů a jejich položek, likvidace faktur
8. Evidence bankovních dokladů - evidence bankovních dokladů a jejich položek, likvidace faktur
9. Evidence zaměstnanců - výpočty mezd, evidence zaměstnanců, mezd a odvodů z mezd
10. Evidence objednávek - evidence objednávek, jejich položek a popisu

11. Evidence nabídek a poptávek - evidence nabídek a poptávek, jejich položek a popisu
12. Sklad - evidence zboží a materiálu na skladu, skladové příjemky a výdejky
13. Peněžní deník - přehled položek a jejich podrobností v peněžním deníku

4.3.2.1 0. úroveň Na této úrovni je vidět pouze rozdělení do modulů.

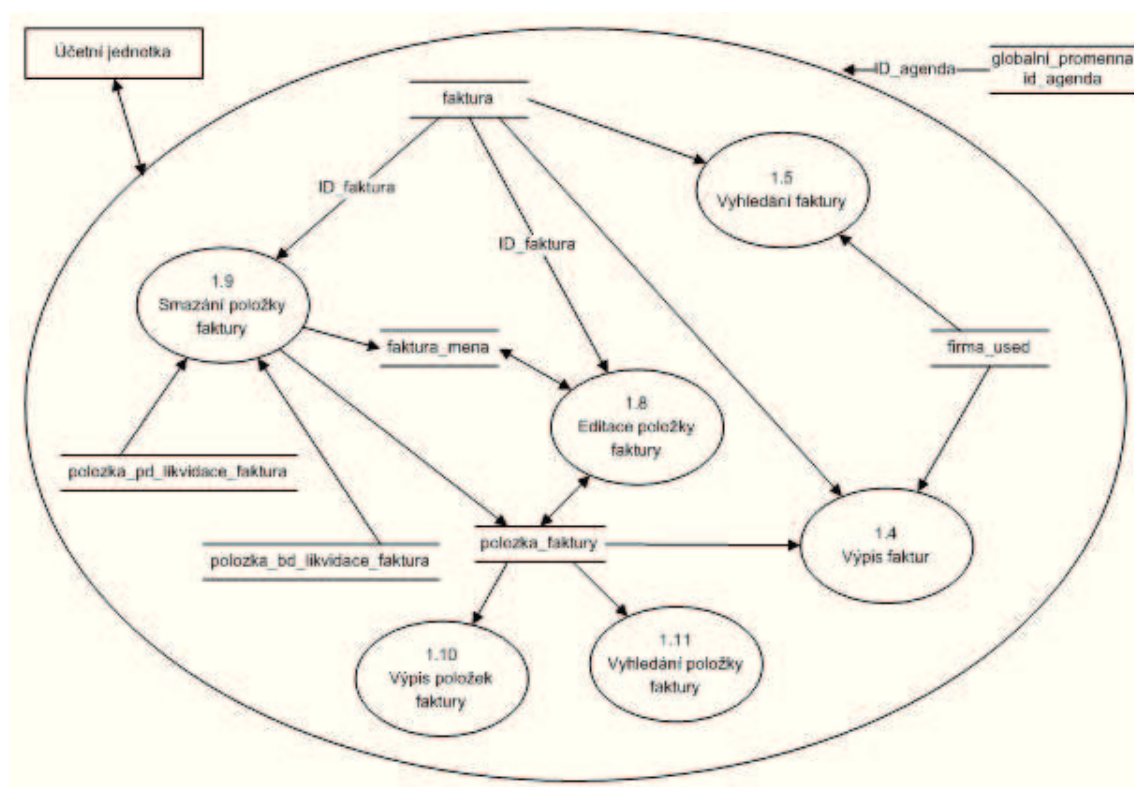
Obrázek 5 na straně 31 znázorňuje informační systém v DFD diagramu 0. úrovně.



Obrázek 5: DFD diagram 0. úrovně

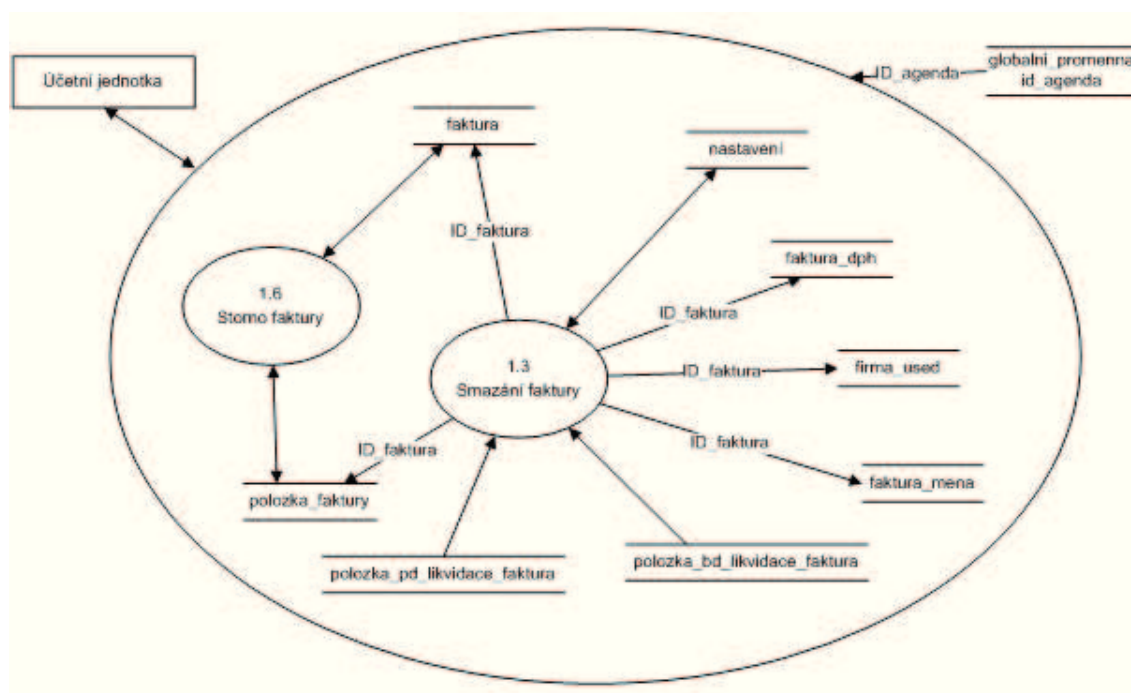
4.3.2.2 1. úroveň Následující diagramy popisují pouze zlomek z celkového počtu zanalyzovaných procesů. Jedná se o vybrané DFD diagramy, týkající se evidence faktur, které podrobněji rozebírají evidenci faktur, uvedenou na 0. úrovni DFD pod číslem procesu 1. Všechny DFD diagramy jsou k dispozici na CD v příloze Funkční analýza. Zbývající diagramy zde nejsou uvedeny kvůli většímu rozsahu části funkční analýzy, zabývající se diagramy datových toků.

1. Evidence faktur I - přidání faktury, editace faktury, přidání položky faktury. Obrázek 6 na straně 32 rozpracovává 1. část evidence faktur do DFD diagramu 1 úrovně.



Obrázek 7: DFD diagram 1. úrovně - evidence faktur II

1. Evidence faktur III - storno faktury, smazání faktury. Obrázek 8 na straně 34 rozpracovává 3. část evidence faktur do DFD diagramu 1 úrovně.

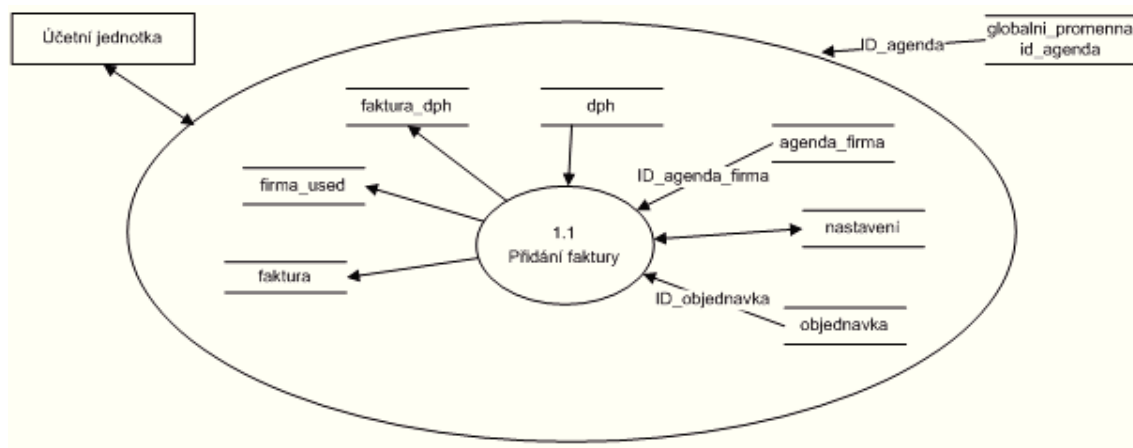


Obrázek 8: DFD diagram 1. úrovně - evidence faktur III

4.3.3 Minispecifikace

Minispecifikace blíže specifikuje jednotlivé procesy. Pro tento účel je u každé minispecifikace uveden daný proces z DFD diagramu 1. úrovně, dále návrh formuláře - tedy první verze uživatelského rozhraní formuláře, určeného pro daný proces a nakonec algoritmus minispecifikace, který má za úkol co nejvhodněji, případně i nejpodrobněji popsat kroky algoritmu, který bude základním vodítkem pro návrh implementace a bude využit také na začátku implementace, kdy bude daný pseudo-algoritmus přepsán do zdrojového kódu.

4.3.3.1 Přidání faktury Na obrázku 9 na straně 35 je uvedena minispecifikace procesu pro přidání faktury.



Obrázek 9: Minispecifikace - přidání faktury

Ukázka formuláře Přidání faktury: Na obrázku 10 na straně 35 je zobrazena ukázka formuláře pro Přidání nové faktury.

The form is divided into two main sections: 'Faktura:' (Invoice) and 'Firma:' (Company).

Faktura:

- Typ: Faktura (dropdown)
- Číslo faktury: 1011000060
- Variabilní symbol:
- Konstantní symbol: 0379 - ostatní příjmy (dropdown)
- Odpočet DPH: Tuzemské plnění (dropdown)
- Předkontace: Prodej zboží (dropdown)
- Forma platby: převodem (dropdown)
- Objednávka č.:
- Sleva: --žádná-- (dropdown)
- (změna slevy se projeví až po uložení faktury)
- Vystavujeme Vám fakturu na zboží dle objednávky
- Text faktury:

Částka v tuzemské měně:

Základní cena	DPH	Celková cena (Kč)
	19	
	9	
	0	

Firma:

- Vyberte firmu ze seznamu: --žádná-- (dropdown)
- Vyberte adresu firmy ze seznamu: --žádná-- (dropdown)
- Vyberte kontakt firmy ze seznamu: --žádný-- (dropdown)
- Název firmy:
- Jméno:
- IČO:
- DIČ:
- Adresa firmy:**
- Ulice:
- Obec:
- PSČ:
- Stát:
- Datum vystavení: 10.02.2011 (calendar icon)
- Datum plnění: 10.02.2011 (calendar icon)
- Datum splatnosti: 24.02.2011 (calendar icon)

Obrázek 10: Ukázka formuláře - přidání faktury

Algoritmus:

1. Přiřaď hodnotu ID_agenda z globální proměnné do paměťové proměnné g.ID_agenda (přiřadí ID přihlášeného uživatele do g.ID_agenda).
2. Vytvoř seznam typů faktury s položkami faktura, dobropis a vrubopis.

3. Načti z tabulky konstantni_symbol do seznamu konstantních symbolů záznamy s atributy ID_konstantni_symbol, nazev a zkratka, kde ID_agenda = g.ID_agenda.
4. Načti z tabulky odpocet_dph do seznamu odpočtů DPH záznamy s atributy ID_odpocet_dph a nazev, kde ID_agenda = g.ID_agenda.
5. Načti z tabulky predkontace do seznamu předkontací záznamy s atributy ID_predkontace a nazev, kde ID_agenda = g.ID_agenda.
6. Načti z tabulky forma_platby do seznamu forem platby záznamy s atributy ID_forma_platby a nazev, kde ID_agenda = g.ID_agenda.
7. Načti z tabulky sleva do seznamu slev záznamy s atributy ID_sleva, nazev a sleva, kde ID_agenda = g.ID_agenda.
8. Načti z tabulky firma do seznamu firem záznamy s atributy ID_firma a nazev, kde ID_agenda = g.ID_agenda.
9. Načti z tabulky nastaveni v závislosti na vybraném typu faktury příslušné atributy s aktuálním číslem (faktury, dobropisu, vrubopisu) a defaultním textem (faktury, dobropisu, vrubopisu). Hodnoty atributů vlož do textových polí cislo_faktury a text.
10. Zobraz formulář Vytvořit (přijatou/vystavenou) fakturu.
11. Uživatel vybere ze seznamu typ faktury. V závislosti na vybraném typu faktury načti příslušné atributy s aktuálním číslem (faktury, dobropisu, vrubopisu) a defaultním textem (faktury, dobropisu, vrubopisu). Hodnoty atributů vlož do textových polí cislo_faktury a text.
12. Uživatel vybere ze seznamu firem firmu.
 - Na základě vybraného záznamu načti z tabulky firma atributy ID_firma, nazev_firmy, ico, dic a přiřaď je do příslušných textových polí nazev_firmy, ico a dic. Uživatel bude moci tyto textové pole dále editovat. Přiřaď atribut ID_firma do paměťové proměnné p.ID_firma.
 - Načti z tabulky adresa_firmy do seznamu adres záznamy s atributy ID_adresa, nazev_adresy, ulice a obec, kde ID_firma = p.ID_firma a zároveň ID_agenda = g.ID_agenda.
 - Načti z tabulky kontakty do seznamu kontaktů záznamy s atributy ID_kontakt, jmeno a prijmeni, kde ID_firma = p.ID_firma a zároveň ID_agenda = g.ID_agenda.
13. Uživatel vybere ze seznamu adres firmy adresu.
 - Na základě vybraného záznamu načti z tabulky adresa_firmy atributy ID_adresa, ulice, obec, psc, stat a přiřaď je do příslušných textových polí ulice, obec, psc, stat. Uživatel bude moci tyto textové pole dále editovat.

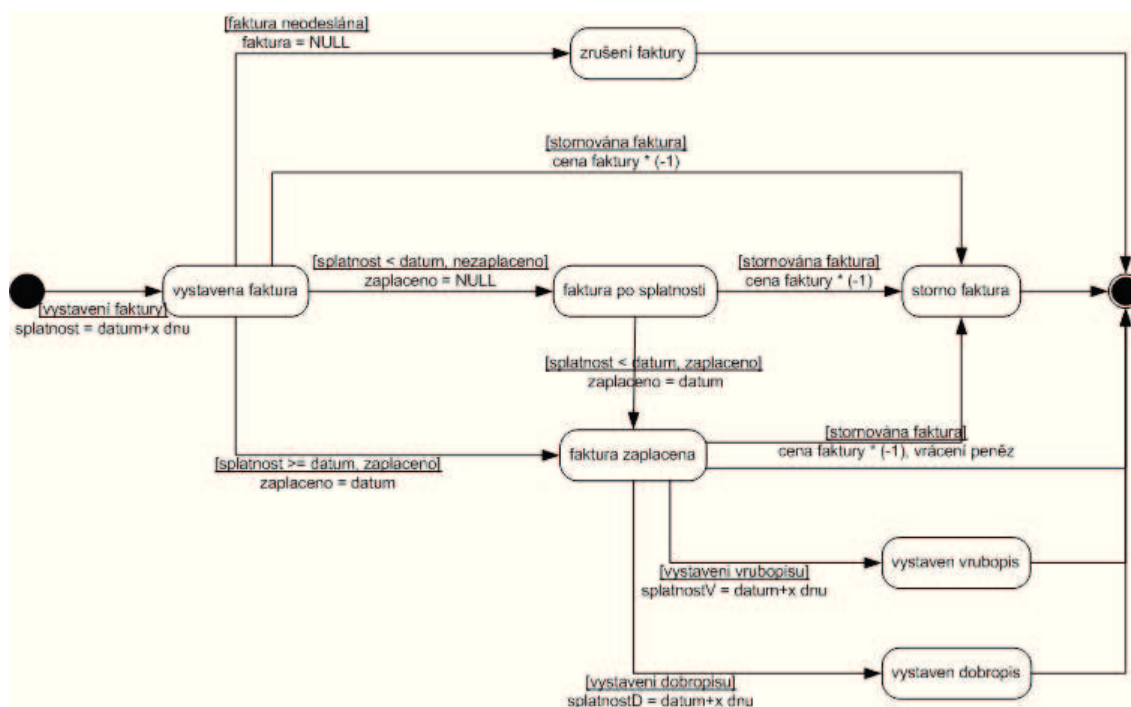
14. Uživatel vybere ze seznamu kontaktů firmy kontakt.
 - Na základě vybraného záznamu načti z tabulky kontakty atributy ID_kontakt, jmeno, prijmeni, spoj atributy jmeno a prijmeni v jeden řetězec (atributy od sebe nejdříve odděl mezerou) a přiřaď jej do textového pole jmeno. Uživatel bude moci toto textové pole dále editovat.
15. Uživatel vyplní a vybere ze zbývajících seznamů všechny povinné údaje do paměťových proměnných formuláře.
16. Zkontroluj, zda byly zadány všechny povinné údaje a zda byly hodnoty do paměťových proměnných zadány ve správném formátu. Jestliže ne, vypiš chybové hlášení „Hodnota pole xxx ve formuláři byla zadána ve špatném formátu.“ a přejdi na krok 10.
17. Číslo faktury ulož do paměťové proměnné p.cislo_faktury.
18. Vyhledej v databázi, v tabulce faktury, zda již číslo faktury p.cislo_faktury existuje a pokud ano, zda také zároveň splňuje podmínku ID_agenda je rovno g.ID_agenda. Pokud existuje, vypiš chybové hlášení „Zadané číslo faktury č. p.cislo_faktury již existuje.“ a přejdi na krok 10.
19. Jestliže uživatel zadal číslo objednávky, ulož jej do paměťové proměnné p.cislo_objednavky.
20. Jestliže je p.cislo_objednavky číslo, vyhledej v databázi, v tabulce objednávky, zda již číslo objednávky p.cislo_objednavky existuje a pokud ano, zda také zároveň splňuje podmínku ID_agenda je rovno g.ID_agenda.
 - a) Pokud takový záznam neexistuje, vypiš chybové hlášení „Zadané číslo objednávky č. p.cislo_objednavky neexistuje.“, do paměťové proměnné p.ID_objednavka ulož NULL a přejdi na krok 10.
 - b) Pokud takový záznam existuje, ulož ID_objednavka vyhledaného záznamu do paměťové proměnné p.ID_objednavka.
21. Načti z databáze, z tabulky nastaveni, vlastnosti zaokrouhlování (zaokrouhlení - na kolik desetinných míst se ceny faktury zaokrouhlí; typ_zakrouhleni - nahoru, dolů, matematicky; zaokrouhleni_polozky - na kolik desetinných míst se zaokrouhlí ceny položek faktury), kde ID_agenda je rovna g.ID_agenda a přiřaď z příslušných proměnných databáze do paměťových proměnných p.zaokrouhleni, p.typ_zakrouhleni, p.zaokrouhleni_polozky.
22. Načti z databáze, z tabulky agenda_firma, data o firmě přihlášené účetní jednotky, kde ID_agenda je rovno g.ID_agenda a ulož je do příslušných paměťových proměnných.
23. Zapiš nový záznam v tabulce firma_used a přiřaď do atributů tabulky hodnoty příslušných paměťových proměnných.

24. Vrat'ID_firma_USED vytvořeného záznamu a ulož do paměťové proměnné p.ID_firma_used.
25. Zapiš nový záznam v tabulce faktura a přiřaď do atributů tabulky hodnoty příslušných paměťových proměnných.
26. Vrat'ID_faktura vytvořeného záznamu a ulož do paměťové proměnné p.ID_faktura.
27. Načti z databáze, z tabulky dph, záznamy s aktuálními sazbami DPH, kde ID_agenda je rovno g.ID_agenda a zapiš nové záznamy v tabulce faktura_dph a přiřaď do atributů tabulky faktura_dph hodnoty příslušných atributů tabulky dph a paměťových proměnných.
28. Jestliže byla uživatelem vyplněna část formuláře pro přidání položky faktury, proved':
Načti z databáze, z tabulky mena, záznamy s aktuálně používanými měnami, kde ID_agenda je rovno g.ID_agenda.
 - Zapiš nové záznamy v tabulce faktura_mena a přiřaď do atributů tabulky faktura_mena hodnoty příslušných atributů tabulky mena a paměťových proměnných.
 - Zapiš nový/é záznam/y v tabulce polozka_faktury a přiřaď do atributů tabulky hodnoty příslušných paměťových proměnných.
29. Uprav aktuální záznam z tabulky nastavení, kde ID_agenda je rovno g.ID_agenda a vlož do příslušného atributu cislo_faktury hodnotu paměťové proměnné p.cislo_faktury+1.
30. Vypiš hlášku „Faktura p.ID_faktura byla vytvořena“.

4.3.4 Dynamická analýza

Dynamická nebo-li časová analýza znázorňuje pomocí stavového diagramu životní cyklus entit z pohledu času. Následující ukázka je pouze výňatkem z kompletní časové analýzy, která není vzhledem k svému rozsahu uvedena přímo v sekci přílohy, ale lze ji najít na přiloženém CD ve složce „přílohy“ v souboru „casova_analyza.pdf“.

Na obrázku 11 uvedeném na straně 39 je STD diagram, graficky znázorňující životní cyklus entity vystavená faktura.



Obrázek 11: STD diagram - vystavená faktura

Upřesnění STD diagramu: pokud je vystavena storno faktura, vždy se musí vztahovat k jiné, než již stornované faktuře. Storno faktura může vzniknout jak po uhrazení faktury, tak před jejím uhrazením a zpravidla tehdy, když již byla odeslána příjemci. V případě uhrazení faktury, je platba vrácena zpět příjemci. Vrubopis je typ faktury, který navyšuje stávající cenu faktury o danou částku. Dobropis je také typ faktury. Jedná se o opak vrubopisu, tedy o slevu z ceny faktury o danou částku. Při rušení faktury se ověřuje z tabulky nastavení, zda aktuální číslo vystavené faktury -1 není rovno číslu rušené faktury. Ověřování probíhá proto, aby nedošlo k porušení číselné posloupnosti vystavených faktur.

4.4 Návrh implementace

Návrh implementace vychází z vytvořené analýzy a ještě podrobněji rozvádí analýzu pomocí vybraných metod návrhu implementace. V návrhu implementace byla provedena transakční analýza pro zpřesnění algoritmů minispecifikací, dále analýza zálohování dat včetně tabulek s periodikami záloh jednotlivých evidencí, analýza archivace dat a nakonec návrh grafického uživatelského rozhraní (GUI) pro vybrané stránky.

4.4.1 Transakční analýza

Databázový systém MySQL [8] obsahuje příkazy pro práci s transakcemi. V případě informačního systému daňové evidence pracuje s databází pouze jedna role a jedna osoba

v podobě uživatele - účetní. Nebývá běžné, aby v jedné agendě konkrétní firmy, která využívá systém účtování pomocí daňové evidence, pracovalo více účetních najednou. Existují však případy, ve kterých je to možné a tedy může i v tomto IS k souběhu dojít - například při vytváření faktur.

Možná ještě větším nebezpečím, kvůli kterému je nutné ošetřit kód pomocí transakcí, může být případná chyba v průběhu zpracování více příkazů v databázi - nedokončené dotazy, které by vedly k neintegritě dat v databázi nebo k pádu informačního systému do neočekávaného stavu.

Transakční analýza slouží k zpřesnění jednotlivých algoritmů minispecifikací o klíčové slova transakcí.

Algoritmus:

1. Přiřaď hodnotu ID_agenda z globální proměnné do paměťové proměnné g.ID_agenda (přiřadí ID přihlášeného uživatele do g.ID_agenda).
2. Vytvoř seznam typů faktury s položkami faktura, dobropis a vrubopis.
3. Načti z tabulky konstantni_symbol do seznamu konstantních symbolů záznamy s atributy ID_konstantni_symbol, nazev a zkratka, kde ID_agenda = g.ID_agenda.
4. Načti z tabulky odpocet_dph do seznamu odpočtů DPH záznamy s atributy ID_odpocet_dph a nazev, kde ID_agenda = g.ID_agenda.
5. Načti z tabulky predkontace do seznamu předkontací záznamy s atributy ID_predkontace a nazev, kde ID_agenda = g.ID_agenda.
6. Načti z tabulky forma_platby do seznamu forem platby záznamy s atributy ID_forma_platby a nazev, kde ID_agenda = g.ID_agenda.
7. Načti z tabulky sleva do seznamu slev záznamy s atributy ID_sleva, nazev a sleva, kde ID_agenda = g.ID_agenda.
8. Načti z tabulky firma do seznamu firem záznamy s atributy ID_firma a nazev, kde ID_agenda = g.ID_agenda.
9. Načti z tabulky nastaveni v závislosti na vybraném typu faktury příslušné atributy s aktuálním číslem (faktury, dobropisu, vrubopisu) a defaultním textem (faktury, dobropisu, vrubopisu). Hodnoty atributů vlož do textových polí cislo_faktury a text.
10. Zobraz formulář Vytvořit (přijatou/vystavenou) fakturu.
11. Uživatel vybere ze seznamu typ faktury. V závislosti na vybraném typu faktury načti příslušné atributy s aktuálním číslem (faktury, dobropisu, vrubopisu) a defaultním textem (faktury, dobropisu, vrubopisu). Hodnoty atributů vlož do textových polí cislo_faktury a text.
12. Uživatel vybere ze seznamu firem firmu.

- Na základě vybraného záznamu načti z tabulky firma atributy ID_firma, nazev_firmy, ico, dic a přiřaď je do příslušných textových polí nazev_firmy, ico a dic. Uživatel bude moci tyto textové pole dále editovat. Přiřaď atribut ID_firma do paměťové proměnné p.ID_firma.
- Načti z tabulky adresa_firmy do seznamu adres záznamy s atributy ID_adresa, nazev_adresy, ulice a obec, kde ID_firma = p.ID_firma a zároveň ID_agenda = g.ID_agenda.
- Načti z tabulky kontakty do seznamu kontaktů záznamy s atributy ID_kontakt, jmeno a prijmeni, kde ID_firma = p.ID_firma a zároveň ID_agenda = g.ID_agenda.

13. Uživatel vybere ze seznamu adres firmy adresu.

- Na základě vybraného záznamu načti z tabulky adresa_firmy atributy ID_adresa, ulice, obec, psc, stat a přiřaď je do příslušných textových polí ulice, obec, psc, stat. Uživatel bude moci tyto textové pole dále editovat.

14. Uživatel vybere ze seznamu kontaktů firmy kontakt.

- Na základě vybraného záznamu načti z tabulky kontakty atributy ID_kontakt, jmeno, prijmeni, spoj atributy jmeno a prijmeni v jeden řetězec (atributy od sebe nejdříve odděl mezerou) a přiřaď jej do textového pole jmeno. Uživatel bude moci toto textové pole dále editovat.

15. Uživatel vyplní a vybere ze zbývajících seznamů všechny povinné údaje do paměťových proměnných formuláře.

16. Zkontroluj, zda byly zadány všechny povinné údaje a zda byly hodnoty do paměťových proměnných zadány ve správném formátu. Jestliže ne, vypiš chybové hlášení „Hodnota pole xxx ve formuláři byla zadána ve špatném formátu.“ a přejdi na krok 10.

17. Číslo faktury ulož do paměťové proměnné p.cislo_faktury.

18. Vyhledej v databázi, v tabulce faktury, zda již číslo faktury p.cislo_faktury existuje a pokud ano, zda také zároveň splňuje podmínku ID_agenda je rovno g.ID_agenda. Pokud existuje, vypiš chybové hlášení „Zadané číslo faktury č. p.cislo_faktury již existuje.“ a přejdi na krok 10.

19. Jestliže uživatel zadal číslo objednávky, ulož jej do paměťové proměnné p.cislo_objednavky.

20. Jestliže je p.cislo_objednavky číslo, vyhledej v databázi, v tabulce objednavky, zda již číslo objednávky p.cislo_objednavky existuje a pokud ano, zda také zároveň splňuje podmínku ID_agenda je rovno g.ID_agenda.

a) Pokud takový záznam neexistuje, vypiš chybové hlášení „Zadané číslo objednávky č. p.cislo_objednavky neexistuje.“, do paměťové proměnné p.ID_objednavka

ulož NULL a přejdi na krok 10.

b) Pokud takový záznam existuje, ulož ID_objednavka vyhledaného záznamu do paměťové proměnné p.ID_objednavka.

21. Načti z databáze, z tabulky nastaveni, vlastnosti zaokrouhlování (zaokrouhlení - na kolik desetinných míst se ceny faktury zaokrouhlí; typ_zakrouhleni - nahoru, dolů, matematicky; zaokrouhleni_polozky - na kolik desetinných míst se zaokrouhlí ceny položek faktury), kde ID_agenda je rovna g.ID_agenda a přiřaď z příslušných proměnných databáze do paměťových proměnných p.zaokrouhleni, p.typ_zakrouhleni, p.zaokrouhleni_polozky.
22. Načti z databáze, z tabulky agenda_firma, data o firmě přihlášené účetní jednotky, kde ID_agenda je rovno g.ID_agenda a ulož je do příslušných paměťových proměnných.
23. BEGIN TRANSACTION
24. Zapiš nový záznam v tabulce firma_used a přiřaď do atributů tabulky hodnoty příslušných paměťových proměnných.
25. Vrať ID_firma_USED vytvořeného záznamu a ulož do paměťové proměnné p.ID_firma_used.
26. Zapiš nový záznam v tabulce faktura a přiřaď do atributů tabulky hodnoty příslušných paměťových proměnných.
27. Vrať ID_faktura vytvořeného záznamu a ulož do paměťové proměnné p.ID_faktura.
28. Načti z databáze, z tabulky dph, záznamy s aktuálními sazbami DPH, kde ID_agenda je rovno g.ID_agenda a zapiš nové záznamy v tabulce faktura_dph a přiřaď do atributů tabulky faktura_dph hodnoty příslušných atributů tabulky dph a paměťových proměnných.
29. Jestliže byla uživatelem vyplněna část formuláře pro přidání položky faktury, proved':
Načti z databáze, z tabulky mena, záznamy s aktuálně používanými měnami, kde ID_agenda je rovno g.ID_agenda.
 - Zapiš nové záznamy v tabulce faktura_mena a přiřaď do atributů tabulky faktura_mena hodnoty příslušných atributů tabulky mena a paměťových proměnných.
 - Zapiš nový záznam v tabulce polozka_faktury a přiřaď do atributů tabulky hodnoty příslušných paměťových proměnných.
30. Uprav aktuální záznam z tabulky nastavení, kde ID_agenda je rovno g.ID_agenda a vlož do příslušného atributu cislo_faktury hodnotu paměťové proměnné p.cislo_faktury+1.

31. END TRANSACTION

32. Vypiš hlášku „Faktura p.ID_faktura byla vytvořena“.

4.4.2 Analýza zálohování dat

Zálohování dat bývá citlivým místem většiny informačních systémů. U větších IS s rozsáhlou databází prochází zálohy složitým procesem plánování. Problém nastává především u serveru, na kterém je informační systém hostován a ve většině případů nebývá server dedikován pouze danému IS, ale poskytuje zároveň více služeb a aplikací (dále procesů), které mohou vyžadovat prostředky serveru (paměť, čas CPU, konektivitu, apod.) v různou dobu vždy s určitou časovou náročností. Proto je nutné plánovat celý proces zálohy dat - databáze, či samotného informačního systému tak, aby svou náročností neomezoval ve vykonávání další procesy.

S tím je velmi úzce spojen problém klientů služeb, kteří potřebují využívat služby bez omezení a pokud je to nevyhnutelné, omezení klienta z časového hlediska minimalizovat. Proto je vhodné plánovat zálohy na dobu, kdy s danými službami nepracují žádní nebo pouze minimum klientů.

Pokud se například jedná o firmu, kde je pracovní doba zaměstnanců, využívajících služby na serveru, v rozmezí od 6:00 do 18:00, je mnoho prostoru pro rozdělení času záloh dat a databází mimo tuto dobu. Problémové je to však u služeb a aplikací, které jsou neustále využívány klienty. V těchto případech je nutné zjistit doby, kdy je počet klientů co nejmenší a provádět zálohu ve vybranou dobu - toto bývá nejčastěji problém veřejných informačních systémů (objednávkové systémy, e-shopy, obecně veřejné weby s proměnlivým obsahem - databází). Aby nedošlo k porušení integrity a návaznosti dat (zejména v zálohách databází), je možné na krátkou dobu uzamknout databázi pro zápis, provést zálohu a po skončení zálohy databázi opět odemknout. Hlavní problémem je dočasné znepřístupnění databáze pro zápis a modifikace, které by mělo proběhnout v co nejkratším čase (s největší prioritou). V případě zamknutí databáze také vzniká nebezpečí možných problémů a chyb, které by znemožnily odemknutí databáze a omezily samotnou službu. Tento způsob se však u většiny malých a středních produkčních řešení příliš nepoužívá.

U informačního systému daňové evidence a jeho nasazení na server v drtivé většině případů stačí naplánovat zálohy na časovou mezeru, kdy není databáze používána vůbec nebo pouze minimálně. Řešením může být použití plánovače úloh (u operačního systému Debian aplikace crontab), pomocí kterého jsou spouštěny zálohy databází (nejčastěji prováděno zálohovacími skripty). Četnost zálohování pak záleží na důležitosti dat, jejich velikosti (velikosti databáze) a časových možnostech provádění záloh. Výsledný interval záloh by měl být nejlepším poměrem mezi uvedenými předpoklady. V následující tabulce 3 na straně 45 je uveden návrh četnosti záloh dat, tedy doporučení pro četnost zálohování jednotlivých tabulek (seznamů a evidencí) databáze.

Návrh četnosti záloh dat - tabulek databáze:

Databázové tabulky	Priorita tabulky	Četnost zálohování
adresa_firmy	3	3 dny
agenda	2	2 dny
agenda_firma	4	5 dní
bankovni_doklad	1	denně
bankovni_doklad_mena	1	denně
bankovni_ucet	1	denně
dph	1	denně
dph_sada	4	5 dní
faktura	1	denně
faktura_dph	1	denně
faktura_mena	1	denně
firma	3	3 dny
firma_used	1	denně
forma_platby	5	týdně
jednotka_mnozstvi	5	týdně
jizda	1	denně
konstantni_symbol	5	týdně
kontakty	3	3 dny
mena	3	3 dny
nabidka_poptavka	1	denně
nabidka_poptavka_mena	1	denně
nastaveni	3	3 dny
objednavka	1	denně
objednavka_mena	1	denně
odpocet_dph	5	týdně
pokladna	1	denně
pokladni_doklad	1	denně
pokladni_doklad_mena	1	denně
polozka_bankovni_doklad	1	denně
polozka_bd_likvidace_faktura	1	denně
polozka_faktury	1	denně
polozka_nabidka_poptavka	1	denně
polozka_objednavka	1	denně
polozka_pd_likvidace_faktura	1	denně
polozka_pokladni_doklad	1	denně
predkontace	5	týdně
ridic	2	2 dny
sleva	5	týdně
vozidlo	2	2 dny
zamestnanec	1	denně
Pokračování na další stránce		

Databázové tabulky	Priorita tabulky	Četnost zálohování
zamestnanec_mzda	1	denně
zamestnanec_odvod	1	denně
zamestnanec_pojistovna	1	denně
zaruka_typ	5	týdně

Tabulka 3: Návrh četnosti záloh dat - tabulek databáze

Poznámka: Atribut Priorita tabulky značí, jakou prioritu mají jednotlivé tabulky, tedy i jejich zálohy (čím nižší číslo, tím vyšší priorita ve stupnici 1-5).

Priority jednotlivých tabulek jsou přímo závislé na četnosti aktualizací záznamů v dané tabulce. Čím čtenější jsou vkládání, editace a mazání záznamů z tabulky, tím vyšší je priorita tabulky pro zálohy.

4.4.3 Analýza archivace dat

Informační systém daňové evidence dokáže pracovat s aktuálními i staršími daty, protože se jedná o účetní program, ve kterém je zapotřebí uchovávat účetní evidenci několik let (podle současného platného zákona minimálně 3 roky, v některých případech až 10 let).

IS také obsahuje výstupní sestavy, které je možné použít pro různé statistiky. Archivace dat tedy není příliš nutná, což samozřejmě neplatí o zálohování, které by, jak již bylo zmíněno v minulé podkapitole, mělo probíhat co nejčastěji, zejména při častých změnách dat v databázi (například před účetní uzávěrkou, na konci měsíce, před daňovým přiznáním, apod.).

4.4.4 Návrh grafického uživatelského rozhraní (GUI)

V současné době ve vývoji webových aplikací a informačních systémů zaujímá grafický design důležitou pozici. Uživatelé si rychle zvykají na čím dál propracovanější grafické detaily v operačních systémech, na webu i graficky orientovaných aplikacích (například hry, grafické modelování, apod.), takže i u informačních systémů, které jsou primárně programovány pro účelnost, začíná být grafické uživatelské rozhraní velmi důležité.

4.4.4.1 Návrh GUI pro evidenci faktur Na následujícím obrázku 12 na stránce 46 lze vidět návrh grafického uživatelského rozhraní pro evidenci vystavených faktur.

Domů Účetnictví Faktury Adresář Zaměstnanci Nastavení Odhlásit se

Vystavené faktury

Faktura:
Typ: Faktura
Číslo faktury: 1011000003
Variabilní symbol: 1011000003
Konstantní symbol: 0379 - ostatní příjmy
Odpočet DPH: Tuzemské plnění
Předkontace: Prodej zboží
Forma platby: hotově
Objednávka č.: 1052000001
Sleva: Sleva 10% - 10.00%
(změna slevy se projeví až po uložení faktury)
Text faktury:

Text faktury

Firma:
Vyberte firmu ze seznamu: --žádná--
Vyberte adresu firmy ze seznamu: --žádná--
Vyberte kontakt firmy ze seznamu: --žádný--
Název firmy: Auston, s.r.o.
Jméno: Jiří Velen
IČO: 54679813
DIČ: CZ54679813
Adresa firmy:
Ulice: Květoslavova 56
Obec: Brno
PSČ: 73300
Stát: ČR
Datum vystavení: 20.03.2010
Datum plnění: 20.03.2010
Datum splatnosti: 22.03.2010

Částka v tuzemské měně:

Základní cena	DPH		Celková cena (Kč)
90	19	17.1	107.1
90	9	8.1	98.1
90	0		295.2

Zobraz záznamů 25

Zobrazují 1 až 25 z celkem 63 záznamů

První Předchozí 1 2 3 Další Poslední

Obrázek 12: Návrh GUI pro evidenci faktur

5 Realizace IS daňové evidence

Informační systém daňové evidence je možné pomyslně rozdělit do jednotlivých modulů, se kterými uživatelé mohou pracovat. Modulem je nejčastěji myšlen konkrétní proces, známý z DFD diagramů, uvedených v části Funkční analýza.

Moduly se mohou dále rozdělit do dvou kategorií podle toho, zda jsou nebo nejsou závislé na jiném modulu. Některé moduly jsou totiž zcela izolovány a lze je brát jako moduly nezávislé na okolí. Nejčastěji se jedná o moduly s jednoduchou funkcionalitou, které mohou využívat další moduly. Složitější moduly tedy částečně využívají funkcionalitu jiných modulů. Z toho se dá zpravidla usoudit, že moduly se složitější funkcionalitou jsou téměř ve všech případech závislé na jednoduchých modulech. V následujících podkapitolách jsou ve zkratce popsány jednotlivé moduly, které budou v rámci IS daňové evidence implementovány.

5.1 Popis modulů IS

Tato část si klade za úkol blíže specifikovat jednotlivé moduly informačního systému, se kterými může uživatel pracovat.

Modul evidence faktur poskytuje uživateli funkce pro práci s fakturami. Jedná se zejména o základní funkce přidání, editace a smazání faktury. Dále může uživatel pracovat s položkami faktury. Položky faktury může opět přidávat, editovat a mazat. Pro likvidaci jednotlivých faktur je nutné využít funkce modulu evidence pokladních dokladů nebo modulu evidence bankovních dokladů.

Modul evidence vozidel využije uživatel, který potřebuje pracovat s evidencemi vozidel, jízdy a řidičů, přičemž u všech těchto evidencí lze pracovat se základními funkcemi přidání, editace a smazání záznamu.

Evidence majetku a odpisy majetku jsou součástí modulu, ve kterém může uživatel přidávat, editovat, odepisovat a mazat drobný, hmotný a nehmotný majetek podle dostupných možností. Odpisy majetku nabízí účetním možnost snížit si základ daně z příjmu.

Evidence firem je jednoduchý modul, který je však hojně využíván mnoha modely pro získání údajů o firmách, jejich adresách a kontaktech. Jedná se tedy v podstatě o adresář firem s pokročilými funkcemi. Uživatel může vytvářet, editovat a mazat firmy, adresy konkrétních firem a kontakty firem. Zatímco adresy musí být přiřazeny ke konkrétní firmě, kontakty nemusí být přiřazeny k žádné firmě nebo mohou být libovolně přemísťovány mezi firmami.

Aby mohl uživatel pracovat s jakýmkoliv dalšími moduly, musí být přihlášen, což obstarává modul přihlášení. Po skončení práce uživatel použije modul odhlášení, který korektně odhlásí uživatele ze systému.

Moduly evidence bankovních a pokladních dokladů poskytují uživateli až na drobné rozdíly stejné funkce. Jedná se především o přidání, editaci a mazání dokladů. V závislosti na příjmovém nebo výdajovém dokladu může uživatel využít funkce likvidace faktury položkou pokladního nebo bankovního dokladu. Samozřejmostí je evidence položek jak

pokladních, tak i bankovních dokladů s funkcemi přidání, editace a mazání jednotlivých položek dokladů.

Modul evidence zaměstnanců obsahuje funkci pro výpočet mezd zaměstnanců za vybraný měsíc, dále běžné funkce pro přidání, editaci a smazání mzdy, zaměstnance a podrobný výpis a editace jednotlivých odvodů ze mzdy zaměstnance.

Moduly evidence objednávek, nabídek a poptávek poskytují funkce pro přidání, editaci a smazání příslušných záznamů k vybraným evidencím včetně přidání, editace a smazání položek objednávek, nabídek či poptávek. Evidence také obsahují možnost vložit ke každému záznamu popis zpřesnit tak evidované položky.

Skladová evidence je robustní modul, obsahující evidenci zboží nebo materiálu dostupného ve skladě. Ve skladě se zboží či materiál eviduje pomocí skladových příjemek a výdejek. Příjemky i výdejky obsahují funkce pro přidání, editaci a smazání zboží či materiálu a svých položek.

Peněžní deník je jednoduchý modul pro chronologický výpis jednotlivých položek evidencí (pokladních a bankovních dokladů, faktur, apod.) do seznamu s potřebnými podrobnostmi všech vypsanych položek. Údaje se do příslušných sloupců peněžního deníku umísťují na základě tzv. předkontakcí, které jsou definovány u položek vypisovaných evidencí.

5.2 Rozbor modulu faktura

Modul faktura je jeden z nejsložitějších a nejzajímavějších modulů celého informačního systému daňové evidence. Využívá mnoho funkcí i jiných modulů. Proto je důležité popsat jednotlivé části modulu dopodrobna tak, aby bylo možné jej kdykoliv upravit, znovu zanalyzovat nebo jej zakomponovat do jiného informačního systému. Tyto části podrobně popisuje zejména datová a funkční analýza. Vzhledem k velkému rozsahu nejsou tyto přílohy uvedeny přímo v sekci přílohy, ale lze je najít na přiloženém CD ve složce „přílohy“ a souborech „datova_analyza.pdf“, „funkcni_analyza.pdf“.

Každá stránka se v základu skládá ze strukturovaného PHP [7] kódu a šablony Smarty, ve které je nakódována stránka v XHTML, přičemž každá stránka je složená z více šablon. Pro všechny stránky jsou společné šablony s hlavičkou a patičkou, kde jsou umístěny základní definice XHTML a externí soubory kaskádových stylů a javaskriptů. Javaskripty [12] v mnoha případech dále používají Ajaxové skripty pro vyřizování asynchronních požadavků, tedy těch, které mění obsah stránky nebo kódu, aniž by se stránka musela znovu načítat.

Stránka pro práci s fakturami se dělí na dvě základní části a to na seznam záznamů (faktur) a detail záznamu (faktury). Seznam faktur je strukturovaný seznam již vytvořených záznamů jednotlivých faktur, který se inicializuje v javascriptu a načítá přes JSON (JavaScript Object Notation) pomocí asynchronního Ajaxového skriptu. Pro vytváření seznamů je použit doplněk pro jQuery s názvem DataTables [5], který obsahuje funkce pro interaktivní práci uživatele se seznamem. Integrované funkce se navíc dají velmi jednoduše konfigurovat. Takovými funkcemi jsou například vzestupné nebo sestupné řazení dle vybraného nebo vybraných atributů, stránkování seznamu dle vybraného počtu záznamů na stranu, vyhledávání v seznamu, apod.

Ukázku stránky pro evidenci faktur lze vidět na obrázku 12 na stránce 46, kde je v horní části stránky umístěn formulář pro vytváření nebo editaci detailu vybrané faktury a ve spodní části seznam existujících faktur.

5.2.1 Interaktivní seznam faktur

Jak již bylo uvedeno, pro načtení seznamu faktur je použit asynchronní Ajaxový skript. Ve skriptu je nejdůležitější SQL dotaz, který načítá z databáze detaily o jednotlivých fakturách, dále ke každé faktuře spočítá celkovou cenu všech položek faktury, spočítá již uhrazenou částku položkami pokladních a bankovních dokladů a pokud je daná faktura evidovaná v cizí měně, přepočítá částky do aktuální tuzemské měny. Pro názornost je zmíněný SQL dotaz uveden v následující ukázce kódu.

```
SELECT SQL_CALC_FOUND_ROWS fa.ID_faktura, fa.ID_agenda, fa.ID_sleva, fa.cislo.faktury, fa.
datum_vystaveni_prijeti, fa.datum_splatnosti, fa.datum_odpocet, fa.text, fa.variabilni_symbol, fa.
prijata_vystavena, fa.typ, fa.ID_firma_USED, fu.nazev_nazevF, fu.jmeno_jmenoF, fu.ulice
uliceF, fu.obec_obecF, fu.psc_pscF, fu.stat statF, fu.ico_icoF, fu.dic_dicF, fu.bu_buF, fu.
kod_bu_kod_buF, s.sleva, pr.ID_predkontace, pr.zkratka_zkratkaPr, pr.nazev_nazevPr, fo.
ID_forma_platby, fo.nazev_nazevFo, ko.ID_konstantni_symbol, ko.nazev_nazevKo, ko.
konstantni_symbol, o.ID_odpocet_dph, obj.cislo_objednavky, fa.zaokrouhleni, fa.
zaokrouhleni_typ, fa.polozky_zobrazeni_zaokrouhleni,
SUM( ( ( ( (d.sazba * 0.01)+1) * pf.zakladni_cena_ks * pf.mnozstvi * ( ( 100 - pf.sleva ) * 0.01
) ) ) *
IFNULL(
( SELECT (fam.kurz/fam.mnozstvi) FROM faktura_mena fam WHERE fam.ID_faktura = fa.
ID_faktura AND typ_meny = 2),
1
)
) AS soucetPolozekFaktury,
( SELECT SUM( ( ( ( (dd.sazba * 0.01)+1) * ppd.zakladni_cena_ks * ppd.mnozstvi * ( ( 100 - ppd.
sleva ) * 0.01 ) ) ) *
IFNULL(
( SELECT (pdm.kurz/pdm.mnozstvi) FROM pokladni_doklad_mena pdm WHERE pdm.
ID_pokladni_doklad = pd.ID_pokladni_doklad AND typ_meny = 2),
1
)
)
) AS soucetPolozekPokladnihoDokladu,
( SELECT SUM( ( ( ( (ddd.sazba * 0.01)+1) * pbd.zakladni_cena_ks * pbd.mnozstvi * ( ( 100 -
pbd.sleva ) * 0.01 ) ) ) *
IFNULL(
( SELECT (bdm.kurz/bdm.mnozstvi) FROM bankovni_doklad_mena bdm WHERE bdm.
ID_bankovni_doklad = bd.ID_bankovni_doklad AND typ_meny = 2),
1
)
```

```

    )
  )
  FROM polozka_bd.likvidace_faktura pbdlf
  LEFT JOIN polozka_bankovni_doklad pbd ON pbd.ID_polozka_bankovni_doklad = pbdlf.
    ID_polozka_bankovni_doklad
  LEFT JOIN bankovni_doklad bd ON bd.ID_bankovni_doklad = pbd.ID_bankovni_doklad
  LEFT JOIN dph ddd ON pbd.ID_dph = ddd.ID_dph
  WHERE fa.ID_faktura = pbdlf.ID_faktura
  GROUP BY fa.ID_faktura
) AS soucetPolozekBankovnihoDokladu

FROM agenda ag, firma_used fu, predkontace pr, forma_platby fo, konstantni_symbol ko,
  odpocet_dph o, faktura fa
LEFT JOIN polozka_faktury pf ON fa.ID_faktura = pf.ID_faktura
LEFT JOIN dph d ON pf.ID_dph = d.ID_dph
LEFT JOIN sleva s ON fa.ID_sleva = s.ID_sleva
LEFT JOIN objednavka obj ON fa.ID_objednavka = obj.ID_objednavka
WHERE fa.ID_agenda = ag.ID_agenda
AND ag.ID_agenda = '$id_agenda'
AND fa.prijata_vystavena = '$prijata_vystavena'
AND fa.ID_firma_USED = fu.ID_firma_USED
AND (fa.typ=1 OR fa.typ=2 OR fa.typ=3)
AND fa.ID_predkontace = pr.ID_predkontace
AND fa.ID_forma_platby = fo.ID_forma_platby
AND fa.ID_konstantni_symbol = ko.ID_konstantni_symbol
AND fa.ID_odpocet_dph = o.ID_odpocet_dph
GROUP BY fa.ID_faktura
$sOrder
$sLimit

```

Výpis 1: SQL dotaz pro načtení seznamu faktur

SQL dotaz spočítá počet nalezených záznamů faktur, které odpovídají daným podmínkám (podmínky uvedeny níže v textu). Tyto podmínky stanovují, jaké záznamy faktur se mají vybrat z databáze. Vybírají se faktury přihlášené agendy s daným ID agendy v proměnné \$id_agenda, dále je určeno, zda se jedná o přijaté nebo vystavené faktury - proměnná \$prijata_vystavena, určuje se výběr konkrétního typu faktury (1 - faktura, 2 - vrubopis, 3 - dobropis). Posledním omezením je maximální počet vybraných záznamů, předaný z prvku DataTables, ve kterém si uživatel může vybrat z přednastavených počtů (proměnná \$sOrder). Proměnná \$sLimit je také předána ze seznamu DataTables podle výchozích nebo uživatelem vybraných seřazení atributů.

Dotaz také spojuje několik tabulek, ze kterých se načítají potřebné proměnné pro uživatelsky přívětivý výpis seznamu. Některé z tabulek, jako například polozka_faktury, dph a jiné, nemusí obsahovat žádné záznamy nebo mohou naopak obsahovat více záznamů, proto je nutné všechny nalezené záznamy faktur „sgrupovat“ dle atributu s ID faktury.

Na začátku dotazu jsou vybírány potřebné atributy z různých tabulek a po nich následuje součet cen všech položek dané faktury, ke které se připočítává sazba DPH dané položky, násobí množstvím výskytů (například počet kusů), pokud existuje, tak je odpočítána od dané položky případná sleva a pokud je faktura navedena do evidence

v cizí měně, je cena přepočítána kurzem z cizí měny do aktuální tuzemské měny pro danou fakturu.

Dále následují dva poddotazy, zvlášť pro pokladní a zvlášť pro bankovní doklady, které podobným způsobem spočítají ceny všech položek všech pokladních a následně i bankovních dokladů, přepočítaných do tuzemské měny, které likvidují (finančně hradí) danou fakturu svými položkami. Položky dokladů mají podobné atributy jako položky faktury. Lze u nich tedy evidovat jednotkovou cenu, množství (počet jednotek, například kusů), k dané položce se vždy vztahuje jedna ze sazeb DPH a je možné u jednotlivých položek evidovat procentuální slevu z ceny položky. Pokladní doklady mohou být vedeny v tuzemské nebo zahraniční měně. To určuje pokladna, ke které se pokladní doklad vztahuje. U bankovních dokladů o měně rozhoduje bankovní účet.

Po načtení seznamu může uživatel dále pracovat se seznamem, řadit záznamy dle jednoho nebo více atributů, procházet stránkováním, určovat počet zobrazených záznamů, atd. V případě faktur nebylo vhodné uživateli zobrazit vyhledávací pole, které v reálném čase filtruje záznamy z databáze na základě uživatelského zadání, protože dotaz je poměrně náročný a při rychlém zadávání znaků uživatelem byla enormně zatížena databáze a docházelo k neuspokojivým časovým odezvám. Seznam faktur byl kvůli delším časovým odezvám, většími, než 1 sekunda, optimalizován pro rychlost, takže nyní dosahuje na stejný počet záznamů v DB více než trojnásobně nižšího času, než tomu bylo před optimalizací dotazu.

5.2.2 Přepočet cen ve formuláři

Při kliknutí uživatelem na vybraný záznam v seznamu faktur, jsou detaily záznamu načteny do formuláře částečně z již načtených hodnot v seznamu a částečně z dalšího Ajaxového skriptu, který rozdělí ceny položek faktury do jednotlivých sazeb DPH, spočítá DPH a celkové ceny k jednotlivým sazbám. Tato část formuláře tedy slouží pro práci s cenami faktury a při vytváření nové faktury může uživatel tyto funkce využít pro přepočet cen ve formuláři. Jedná se o javascriptové funkce, které dopočítají zbývající ceny ve formuláři dle uživatelem zadaných hodnot k jednotlivým sazbám DPH. Ceny k jednotlivým sazbám se přepočítávají při změně některého ze vstupních polí pro základní nebo celkovou cenu. Zároveň je vždy upraven součet ze všech celkových cen ve formuláři. Po vytvoření faktury jsou z této části formuláře na základě zadaných cen (pokud tuto možnost uživatel využil) vytvořeny položky faktury, přiřazené příslušným sazbám DPH.

Na následujícím obrázku 13 na straně 52 lze vidět část formuláře pro přepočítání cen faktury.

Částka v tuzemské měně:

Základní cena	DPH		Celková cena (Kč)
100	20	20	120
120	10	12	132
150	0		402

Obrázek 13: Část formuláře pro přepočet cen faktury

Při vytváření faktury v cizí měně se uživateli tato část formuláře přizpůsobí pro práci s cizí měnou. Zmizí pole pro výběr DPH a objeví se pole pro zadání kurzu a množství kurzu, rolovací seznam pro výběr měny, pole s tuzemskou celkovou cenou a pole s cenou v cizí měně.

5.2.3 Generování tiskových sestav do PDF

Při vytváření informačního systému pro vedení účetnictví pomocí systému daňové evidence je nutné používat tiskové sestavy. Jedná se například o výplatní pásky, různé dokumenty, peněžní deník a v neposlední řadě tisk vystavených faktur. Vzhledem k použití webového rozhraní není příliš vhodné navrhovat tiskové sestavy pro přímý tisk z prohlížeče z hlediska možné nekompatibility některých prohlížečů a s tím spojených dalších problémů - nutnost vytvářet nový soubor s kaskádovými styly speciálně pro tisk a jiné.

Právě tyto argumenty byly jedny z hlavních důvodů, proč vyřešit problém tiskových sestav elegantněji, efektivněji a také pohodlněji. Výsledkem byl požadavek, navrhnout tiskové sestavy tak, aby se exportovaly do PDF, čímž byl zároveň zajištěn i náhled pro uživatele před samotným tiskem dané sestavy. Dalšími výhodami je také možnost vygenerovanou sestavu přímo uložit ve formátu PDF do souboru, PDF soubory dále archivovat, rozesílat elektronickou poštou nebo nahrávat do případných dalších informačních systémů nebo aplikací, jakými může být například CRM, apod.

Samozřejmostí a také primárním účelem vygenerované sestavy je samotný tisk, který zejména v oblasti účetnictví slouží primárně k archivaci sestav, aby bylo možné při případné kontrole finančním úřadem nebo jinými orgány, prokázat, že daná sestava (dokument, faktura, apod.) skutečně existuje a je řádně potvrzena - podpisem, razítkem.

Existuje řada možností a nástrojů, jakými lze vytvořit PDF soubor, většina z nich je však mnohdy pro tento účel nedostatečně vybavena, například některé generátory nejsou zcela funkční nebo nemají příliš mnoho možností, jakými lze stránku stylovat anebo jsou příliš složité na to, aby se pomocí nich mohly vytvářet efektivně složitěji členěné PDF soubory. Typickým příkladem složitějšího nástroje pro vytváření PDF souborů je integrovaná knihovna PHP pro práci s PDF soubory. Mým cílem tedy bylo najít co nejjednodušší generátor PDF souborů, ideálně z HTML kódu, který by byl zároveň napsaný v PHP. Volba padla na velmi často chválený PDF generátor mPDF[14].

Generátor mPDF je třída pro generování PDF souborů z HTML kódu, případně PHP kódu. Při generování vystavených faktur jsou pomocí jednoduše nastavitelných parametrů stanoveny okraje výsledného PDF souboru, patička s místem pro podpisy, razítko a případnou poznámku a hlavička s povinnými atributy faktury, jakými jsou název dodavatelské a odběratelské firmy, jejich adresy a další nutné náležitosti faktury. Proměnlivým obsahem je soupis položek faktury cenami a následnou tabulkou s rozpisem sazeb DPH. Pokud by se stalo, že počet položek bude vyšší a výsledný text bude přesahovat hranice stránky, vytvoří se automaticky nová stránka se zbývajícím textem a znovu vyplněnou hlavičkou a patičkou. Jednou z nevýhod mPDF generátoru je problém se stylováním stránky, protože mnohdy nezobrazuje zcela validně všechny prvky kaskádových stylů CSS a je nutné v tomto směru hodně improvizovat a vrátit se k velmi zastaralému tabulkovému layoutu stránky.

Na následujícím obrázku 14 na straně 54 je ukázka již vygenerované vystavené faktury do PDF pomocí generátoru mPDF.

Faktura - daňový doklad č. 1011000003

<p><small>Dodavatel:</small></p> <p>Ajax s.r.o. Bezručova 560 75301 Hranice</p> <p style="text-align: center;"></p> <p> <small>IČ: 97643113 DIČ: CZ97643113</small> <small>Krajský obchodní soud v Olomouci, oddíl C, vložka 31698</small> <small>tel: +420581603274 e-mail: email@firma.cz</small> <small>fax: +420581603275 web: www.firma.cz</small> </p>	<p><small>Odběratel:</small></p> <p>Auston, s.r.o. Jiří Velen Květoslavova 56 73300 Brno ČR</p> <p> <small>IČ: 54679813 DIČ: CZ54679813</small> </p>
--	--

Forma úhrady:	hotově	Datum vystavení:	20.03.2010
Bankovní spojení:	ČSOB a.s.	Datum zdanitelného plnění:	20.03.2010
Číslo účtu:	199131010/0300	Datum splatnosti:	22.03.2010
Variabilní symbol:	1011000003	Číslo objednávky:	
Konstantní symbol:	0379	Vystavil:	Jiří Král

Text faktury

Název	Množství	Cena / ks	Sleva %	Cena	Sazba DPH	Cena vč. DPH
Položka č.1	1.00 ks	100 Kč	10%	90 Kč	17.1%	107.1 Kč
Položka č. 2	1.00 ks	100 Kč	10%	90 Kč	8.1%	98.1 Kč
Položka č. 3	1.00 ks	100 Kč	10%	90 Kč	0%	90 Kč
Součet položek				270 Kč	25.2 Kč	295.2 Kč
Zaokrouhlení						0 Kč
CELKEM K ÚHRADĚ						295.2 Kč

Viditelný popis faktury...

Rekapitulace DPH		
Sazba DPH	Základ daně	DPH
0%	90 Kč	0 Kč
9%	90 Kč	8.1 Kč
19%	90 Kč	17.1 Kč

Vystavil

Převzal

Obrázek 14: Ukázka vygenerované vystavené faktury do PDF souboru

6 Protokol testování

V dnešní době existuje stále větší množství moderních trendů, které zasahují do vývoje softwaru. Jedná se o doplňky a knihovny, jako například jQuery, ale také komponentní technologie, kterou využívá například .NET. Software začíná být, s ohledem na čím dál vyšší nároky uživatele, stále komplikovanější, takže disciplína vývoje - testování - je již nedílnou součástí vývoje a v mnoha případech se již nebere na lehkou váhu, tak jako tomu mohlo být v minulosti. Ne jinak tomu je i v oboru informačních systémů a webových aplikací. Problémem však zůstává otázka, jak testovat, aby výsledný produkt měl co nejméně chyb a zároveň co nejvyšší jakost. Obecně platí čím více testů, kterými software musí projít, tím je jeho finální jakost vyšší. V praxi se však volí kompromis mezi náklady na testování a výslednou jakostí produktu.

6.1 Testování informačního systému

Podobný případ kompromisu nastal i u testování vyvíjeného informačního systému daňové evidence. Ten však očekávaně nastal mezi časovou dotací a výslednou jakostí produktu. Testovaný systém tedy prošel několika testy a uživatel s ním může pracovat. Tak jako každý SW ovšem stále obsahuje chyby, ať už skryté, na které se při testování nepřišlo, nebo chyby, které se v průběhu vývoje průběžně opravují.

Informační systém jako celek i jednotlivé strany byly podrobeny několika následujícím testům. Je nutné upozornit, že nebyly naimplementovány a provedeny jednotkové testy (například PHPUnit), které by automatizovaně a zejména stejným způsobem testovaly všechny stránky, na základě jejichž výsledků by byly zjištěny případné problémy a přesně spočítány časové odezvy jednotlivých stránek.

Test. disciplína	Výsledek	Podrobnější popis
Testování funkcionality	OK	Ověření správné funkčnosti formulářových prvků, skriptů, výpočtů vkládáním hodnot a běžnou uživatelskou kontrolou výsledných hodnot
Testování bezpečnosti formulářových prvků	OK	Testování zabezpečení GET a POST proměnných proti SQL injections a XSS útokům.
Testování bezpečnosti GET a POST proměnných	Upozornění - nebyly testovány všechny strany - ne všechny případy výskytů GET a POST proměnných jsou proti nebezpečí napadení zabezpečeny.	Testování zabezpečení GET a POST proměnných proti SQL injections a XSS útokům.
Pokračování na další stránce		

Test. disciplína	Výsledek	Podrobnější popis
Testování správnosti validace hodnot formulářů	OK - všechny povinné i nepovinné formulářové prvky byly korektně otestovány.	Ověření správnosti všech validací formulářových prvků pomocí JS.
Testování XHTML validity	Upozornění - 1 ze stran neprošla prováděným testem (zamestnanec-mzda.php)	Validita stran dle konceptu XHTML 1.0 Transitional - Mozilla Firefox validátor XHTML 0.8.6.1
Zátěžový test aplikace	Neprovedeno - test nebyl proveden	Zachování funkcionality a optimalizace aplikace při zátěži
Zátěžový test práce aplikace s databází	OK	Optimalizovány složité SQL dotazy, SQL dotazy otestovány v zátěži, zabezpečení real-time vyhledávání v databázi v seznamech (prvku DataTables) proti přetížení

Tabulka 4: Testování informačního systému

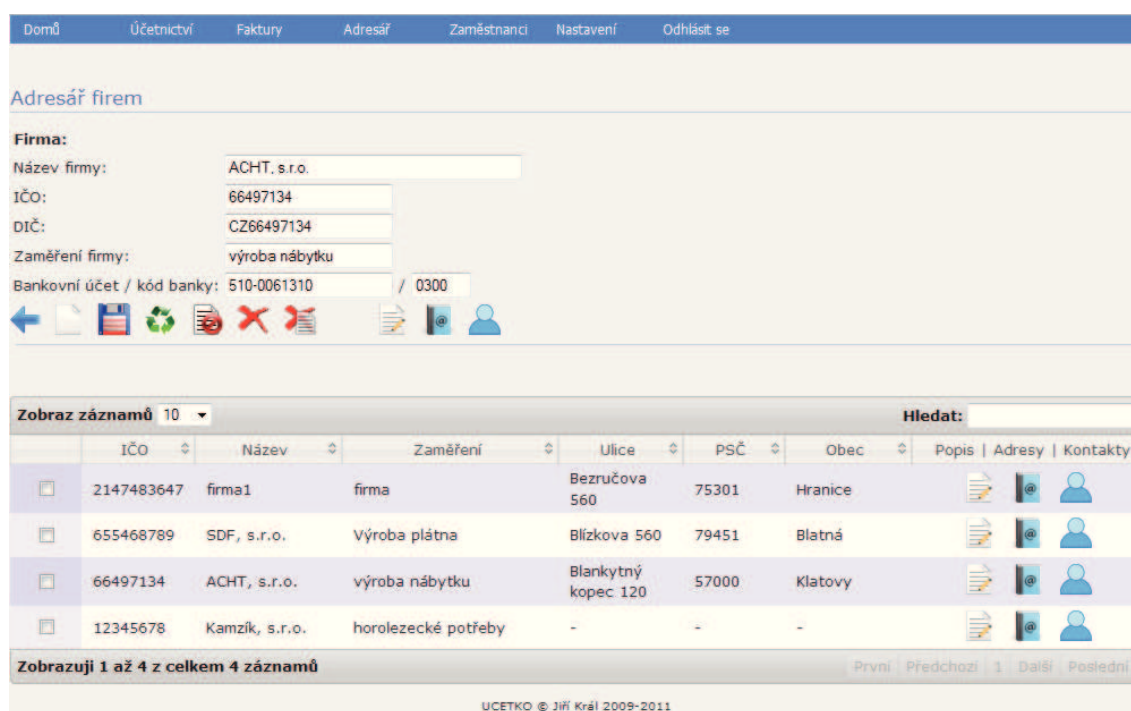
7 Ukázkové příklady pro výuku

Před vybráním ukázkového příkladu je vhodné zamyslet se nad cílovou skupinou studentů, pro které budou ukázky určeny. Jedná-li se o studenty, kteří zatím nemají příliš zkušeností s analýzou a realizací informačních systémů, je lépe nejprve začínat na jednodušších ukázkách, v tomto případě modulech a postupem času a přibývajících zkušeností studentů se propracovávat ke složitějším modulům, které již vyžadují určité znalosti v oboru databázových a informačních systémů.

Řešení složitých příkladů na počátku výuky předmětu mají na většinu studentů dle mého názoru spíše demotivační účinky a úbytek snahy učit se novým věcem, které se v danou chvíli studentům jeví jako složité nebo těžko naučitelné. Tato teze by však mohla být spíše předmětem řešení v oboru psychologie a není nutné ji zde dále rozvádět.

Záleží tedy na každém vyučujícím, kterou část informačního systému použije pro výuku jako názorný příklad. Z mého pohledu se jeví jako nejvhodnější ukázka modul adresáře, který v sobě obsahuje evidenci firem, adres konkrétních firem a kontaktů. Modul je zcela nezávislý na ostatních modulech a tvoří tedy samostatný funkční celek. Jedná se o adresář, jehož význam je obecně známý a bude tedy snazší vysvětlit i jeho funkcionalitu. Studenti pak budou moci sami analyzovat, navrhovat a implementovat části adresáře a pro vyučující bude jednodušší jednotlivé části kontrolovat a případně i hodnotit.

Na obrázku 15 na straně 57 je zobrazena ukázka grafického uživatelského rozhraní pro část modulu adresáře s názvem evidence firem.



Obrázek 15: Náhled na GUI části adresáře - evidence firem

Pro pokročilejší výuku je vhodné vybrat některé názorné, třeba i složitější ukázky analýzy a kódu a ty pak popsat. Studenti tak budou mít možnost shlédnout pokročilejší možnosti analýzy, kódu, relačních databází a později je mít k dispozici jako inspiraci pro vypracování svých projektů nebo k dalšímu vzdělávání v daném oboru. Pro tento účel je vhodný právě modul již mnohokrát zmiňované evidence faktur, který již obsahuje pokročilejší analýzu, kód, skripty a výběry dat z databáze.

Tato podkapitola je však pouze subjektivním názorem autora a záleží na každém vyučujícím, zda využije těchto doporučení nebo zda si výuku přizpůsobí dle svého vlastního uvážení. Na CD je k dispozici příloha se zdrojovými kódy samostatného modulu evidence firem - adresáře se všemi svými funkcemi.

8 Závěr

Cílem diplomové práce bylo vytvořit informační systém daňové evidence, který by byl použitelný v menších firmách, které vedou účetní agendu systémem daňové evidence. Tvorba informačního systému měla probíhat od vytvoření zadání a specifikace požadavků, přes analýzu a návrh implementace až k implementaci samotného informačního systému, tedy všech součástí potřebných pro korektní vedení účetní agendy formou daňové evidence. Naimplementovaný informační systém měl být otestován a část systému měla být zpracována jako ukázkový příklad do databázových předmětů pro studenty navazujícího studia.

Na počátku bylo v zadání stanoveno několik konkrétních požadavků na informační systém, které měly být splněny. Jednalo se zejména o jmenované evidence, z nichž byly všechny s výjimkou skladové evidence zanalyzovány a naimplementovány. Vzhledem k většímu rozsahu diplomové práce a nadstandardních požadavků na nový informační systém daňové evidence, není zakomponování skladové evidence zcela nezbytné. Většina účetních programů v základních verzích, určených pro účtování systémem daňové evidence rovněž nemají obsažen modul pro skladovou evidenci a pro běžné účely drtivé většiny malých firem postačují zbývající moduly, z nichž některé již implementované v informačním systému, jsou také v porovnání k základním verzím spíše nadstandardem, obsaženým v dražších verzích účetních programů. Byla také vypracována část informačního systému jako vzorový příklad do databázových předmětů navazujícího studia. Speciálně pro tento účel byl vybrán modul evidence firem - adresář, který je znám v mnoha podobách široké veřejnosti a tedy i významem běžně srozumitelný.

Za největší přínos diplomové práce, považuji zejména pokročilou funkcionalitu evidence faktur, která interaktivně přepočítává ceny, pracuje s mnoha seznamy, zobrazuje uživateli stavy jednotlivých faktur v seznamu faktur, pracuje korektně s více měnami, případně i souběžnou měnou - připraveno na účtování v souběžné měně, například při přechodu měny na Euro, korektně zaokrouhluje částky jednotlivých faktur a jejich položek v závislosti na definovaném nastavení.

Dalším zajímavým modulem je podle mého názoru evidence zaměstnanců a mezd. Evidence mezd zaměstnanců totiž používá pokročilé výpočty mezd a odvodů z mezd, zahrnující odpočty nezdaniitelných částek, dovolenou a nemocenskou zaměstnance a další možné atributy, ovlivňující výši mzdy a výše odvodů ze mzdy zaměstnance. Evidence také obsahuje správu odvodů mezd, kde má uživatel informačního systému přehled o zaplacených či nezaplacených mzdách a odvodech z mezd.

V budoucnu bych velmi rád dále rozšířil funkcionalitu informačního systému o již zmiňovanou skladovou evidenci a dále o další možnosti nastavení parametrů evidencí tak, aby uživatel mohl měnit téměř všechny parametry a nastavení programu sám nebo s minimální podporou administrátora informačního systému a informační systém by tak pružněji reagoval na možné změny účetních zákonů. To je však asi věčným problémem všech účetních programů, které tak musí procházet četnými změnami na začátku, na konci nebo také v průběhu účetního roku. Bylo by však velmi příjemné vytvořit IS, který by vyžadoval aktualizace v co možná nejdelších časových intervalech, aby tak

poskytl uživateli větší možnosti a neuvěznil ho ve stálém koloběhu plateb za aktualizace a řešeních problémů s nimi spojených.

Dále by bylo vhodné provést důkladnější testování pomocí nástrojů pro testování a automatizovaných testů PHP a javaskriptového kódu, popřípadě i otestovat chování informačního systému při zadávání extrémních hodnot a jeho stabilitu při větší generované zátěži.

Zároveň je nanejvýš nutné důkladněji otestovat zabezpečení informačního systému a provést další kroky ke zvýšení bezpečnosti proti potenciálním útočníkům a to zejména v případě nasazení informačního systému na veřejný server, kde se riziko napadení, nežádoucích zásahů do databáze a případného zneužití citlivých dat, obsažených v účetních evidencích, enormně zvyšuje.

9 Reference

- [1] Internet Info, s.r.o. *Podnikatel.cz* [online]. 2007 [cit. 2011-04-15]. Podnikatel.cz. Dostupné z WWW: <<http://www.podnikatel.cz/dane/ucetnictvi-danovaevidence/co-je-ucetnictvi/>>. ISSN 1802-8012.
- [2] BAVOR, Bc. Jan. *Janbavor.cz* [online]. 2010 [cit. 2011-04-15]. BAVOR bookkeeping. Dostupné z WWW: <http://janbavor.cz/index.php?option=com_content&view=article&id=2&Itemid=8>.
- [3] GOLA, Petr. *Finance.cz* [online]. 2010 [cit. 2011-04-15]. Finance.cz. Dostupné z WWW: <<http://www.finance.cz/zpravy/finance/265313-jake-udaje-musi-obsahovat-danova-evidence-/>>.
- [4] Oracle Corporation and/or its affiliates. *Mysql.com* [online]. 2010 [cit. 2011-04-15]. MySQL Workbench (GUI Tool). Dostupné z WWW: <<http://dev.mysql.com/downloads/workbench/5.2.html>>.
- [5] JARDINE, Allan. *DataTables (table plug-in for jQuery)* [online]. 2007 [cit. 2011-04-15]. Dostupné z WWW: <<http://www.datatables.net/>>.
- [6] ŠARMANOVÁ, Jana. *Databázové a informační systémy*. Ostrava : Vysoká Škola Báňská - Technická univerzita Ostrava, 2007. 122 s. Učební text. ISBN 978-80-248-1499-5.
- [7] The PHP Group. *PHP: Hypertext Preprocessor* [online]. 2001 [cit. 2011-04-16]. Dostupné z WWW: <<http://www.php.net/>>.
- [8] Oracle Corporation and/or its affiliates. *MySQL - The world's most popular open source database* [online]. 2010 [cit. 2011-04-16]. Dostupné z WWW: <<http://www.mysql.com/>>.
- [9] W3C. *XHTML 1.0: The Extensible HyperText Markup Language (Second Edition)* [online]. 2002 [cit. 2011-05-02]. Dostupné z WWW: <<http://www.w3.org/TR/xhtml1/>>.
- [10] New Digital Group, Inc. *PHP Template Engine - Smarty* [online]. 2002 [cit. 2011-04-16]. Dostupné z WWW: <<http://www.smarty.net/>>.
- [11] Refsnes Data. *W3Schools Online Web Tutorials - CSS* [online]. 1999 [cit. 2011-04-16]. Dostupné z WWW: <<http://www.w3schools.com/css/>>.
- [12] W3C. *JavaScript Tutorial* [online]. 1999 [cit. 2011-04-16]. Dostupné z WWW: <<http://www.w3schools.com/js/default.asp>>.
- [13] The jQuery Project. *jQuery - The Write Less, Do More, JavaScript Library* [online]. 2010 [cit. 2011-04-16]. Dostupné z WWW: <<http://jquery.com/>>.
- [14] BACK, Ian. *MPDF* [online]. 2008 [cit. 2011-04-17]. Dostupné z WWW: <<http://www.mpdf1.com/mpdf/>>.

- [15] CÍGLER SOFTWARE, a.s. *CÍGLER SOFTWARE* [online]. 2010 [cit. 2011-04-25]. Účetní program zdarma: Money S3 Start. Dostupné z WWW: <<http://www.money.cz/kestazeni/start>>.
- [16] STORMWARE s.r.o. *POHODA - Ekonomický systém* [online]. 2011 [cit. 2011-04-25]. Účetní software zdarma. Dostupné z WWW: <<http://www.stormware.cz/kestazeni/>>.

10 Obsah CD

V následujícím seznamu je popsáno rozdělení obsahu CD do jednotlivých složek.

- **diplomova_prace**
Ve složce „diplomova_prace“ se nachází hlavní text diplomové práce ve formátu PDF a v podsložce „latex“ zdrojové soubory \LaTeX pro další možnou editaci textu. Ve složce „diplomova_prace“ je také uloženo zadání diplomové práce v souboru „zadani.pdf“.
- **prilohy**
V této složce jsou umístěny veškeré přílohy k této diplomové práci. Jedná se o následující přílohy (v závorce je uveden počet stran přílohy):
 - Specifikace požadavků - soubor „specifikace_pozadavku.pdf“ (10).
 - Datová analýza - soubor „datova_analyza.pdf“ (29).
 - Funkční analýza - soubor „funkcni_analyza.pdf“ (103).
 - Časová analýza - soubor „casova_analyza.pdf“ (5).
 - Návrh implementace - soubor „navrh_implementace.pdf“ (24).
 - Protokol testování - soubor „protokol_testovani.pdf“ (3).
 - Nasazení aplikace - soubor „nasazeni_aplikace.pdf“ (3).
 - Uživatelská příručka - soubor „uzivatelska_prirucka.pdf“ (46).
 - Programátorská příručka - složka „programatorska_prirucka“, ve které je vygenerována dokumentace. Pro spuštění je nutné otevřít v prohlížeči nebo pomocí prohlížeče soubor „index.html“.
- **zdrojove_kody**
Ve složce „zdrojove_kody“ jsou umístěny veškeré zdrojové kódy k informačnímu systému daňové evidence, včetně databáze, umístěné v souboru „ucetnictvi.sql“.
- **zdrojove_kody_adresar**
Ve složce „zdrojove_kody_adresar“ jsou umístěny zdrojové kódy k samostatnému modulu informačního systému, určeného pro výuku databázových předmětů, kterým je evidence firem - adresář.